

Android aplikace pro měření vodivosti pokožky

Android Application for Galvanic Skin Response Measurement

Jakub Válek

Bakalářská práce

Vedoucí práce: Ing. Lukáš Tomaszek

Ostrava, 2021

Abstrakt

Tato bakalářská práce se zabývá vývojem aplikace pro měření vodivosti pokožky využitím mobilního zařízení na platformě Android. Aplikace se bude umět připojit na měřicí zařízení Sensetio za pomoci Bluetooth a získávat z něho potřebné data. Získané data poté budou uchovávána na mobilním zařízení s možností exportu a bude na nich provedena základní analýza. Aplikace by měla poskytnout více možností měření a to s (např. přehrávání videa) nebo bez stimulů. Získané měření pak bude možno filtrovat pomocí různých parametrů a zpětně je zobrazit. Společně s aplikací bude napsána příručka, která bude vysvětlovat práci s aplikací. Aplikace pak bude otestována na několika mobilních zařízeních s jinými verzemi systému Android.

Klíčová slova

Vývoj aplikace; galvanická kožní reakce; vodivost pokožky; elektrodermální aktivita; mobilní zařízení; Sensetio; Android; Bluetooth

Abstract

This bachelor thesis deals with the development of an application for measuring skin conductivity using a mobile device on the Android platform. The application will be able to connect to the Sensetio measuring device via Bluetooth and obtain the necessary data from it. The obtained data will be stored on the mobile device with the possibility of exporting the data and performing a basic analysis with the data. The application should provide more measurement options with (e.g. video playback) or without stimuli. The obtained measurements can then be filtered using different parameters and displayed retrospectively. Along with the application, a manual will be written explaining how to work with the application. The application will then be tested on several mobile devices running on different versions of the Android system.

Keywords

App development; galvanic skin response; skin conductance; electrodermal activity; mobile devices; Sensetio; Android; Bluetooth

Poděkování

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla. Hlavně vedoucímu této bakalářské práce Ing. Lukáši Tomaszkovi za rady a rychlou odezvu vždy, když jsem ji potřeboval.

Obsah

Seznam použitých symbolů a zkratek	7
Seznam obrázků	8
Seznam tabulek	9
1 Úvod	10
2 Kůže a pocení	12
2.1 Anatomie kůže	12
2.2 Pocení	13
3 Principy elektrodermálních jevů	15
3.1 Historie elektrodermální aktivity	15
3.2 Terminologie	15
4 Android	18
4.1 Historie	18
4.2 Architektura	19
4.3 Komponenty	21
4.4 Vlastnosti OS Android	21
5 Použité technologie, vývojové nástroje a prostředí	25
5.1 Android studio	25
5.2 JDK	25
5.3 Android software development kit (SDK)	26
5.4 AVD	26
5.5 XML	26
5.6 CSV	26
5.7 BLE	26
5.8 Zařízení Sensatio	28

6	Návrh	29
6.1	Účel práce	29
6.2	Požadované funkce aplikace	29
6.3	Dodatečné funkce aplikace	30
6.4	Výběr programovacího jazyka a prostředí	32
6.5	Jazyk aplikace	32
6.6	Verze androidu	32
6.7	Orientace mobilu	32
6.8	Profily v aplikaci	32
6.9	Uspořádání dat a úložiště	33
6.10	Vizualizace dat	33
6.11	Vizualizace měření	33
6.12	Komunikace se Sensatio zařízením	34
6.13	Návrh UI	34
7	Implementace	36
7.1	Globalizace	36
7.2	První spuštění aplikace	36
7.3	Navigace	37
7.4	ViewModely a změna orientace mobilu	38
7.5	Bluetooth	38
7.6	Umístění dat ve fyzickém úložišti mobilního zařízení	41
7.7	Zpětná vazba	41
7.8	Nastavení	42
7.9	Hlavní stránka	42
7.10	Vizualizace pomocí grafů GraphView	43
7.11	Měření	43
7.12	Provedení analýzy na datech	46
7.13	Profily	47
7.14	Přehled měření a analýz	50
7.15	O aplikaci	51
8	Testování aplikace	53
8.1	Testování aplikace během vývoje	53
8.2	Experimentální část	53
9	Závěr	56
	Literatura	57

Přílohy	59
A Obsah přílohy	60
A.1 Zdrojový kód aplikace	60
A.2 Instalační soubor .apk	60
B Příručka	61
B.1 Tvorba profilu	61
B.2 Připojení zařízení	62
B.3 Připojení zařízení	65
B.4 Export dat z daného profilu	67
B.5 Přehled měření	68
C Sekvenční diagram připojení bluetooth	70

Seznam použitých zkratek a symbolů

GSR	– Galvanická kožní reakce
EDA	– Elektrodermální aktivita
SP	– Kožní potenciál
SPL	– Úroveň kožního potenciálu
SPR	– Odezva kožního potenciálu
SC	– Kožní vodivost
SCL	– Úroveň kožní vodivosti
SCR	– Odezva kožní vodivosti
SR	– Kožní odpor
UI	– Uživatelské rozhraní
OS	– Operační systém
AVD	– Virtuální zařízení android
CSV	– Čárkou oddělené hodnoty
BLE	– Bluetooth s nízkou spotřebou energie
Inc	– Společnost s ručením omezeným
API	– Rozhraní pro programování aplikací

Seznam obrázků

2.1	Struktura kůže [10]	14
4.1	Architektura operačního systému Android [18]	19
4.2	Ilustrace životního cyklu aktivit [20]	24
5.1	Náramek Sensatio [3]	28
6.1	Use case aplikace	31
7.1	Vzhled příručky v aplikaci	37
7.2	Toolbar a rozbalené menu	38
7.3	Vzhled Bluetooth aktivity	42
7.4	Vzhled stránky s nastavením	43
7.5	Vzhled hlavní stránky s možností rychlého připojení	44
7.6	Vzhled tvoření profilu a stránky s profily	47
7.7	Vzhled aktivity přehled profilu a dialogu pro filtrování měření	48
7.8	Přehledy měření a analýz	52
7.9	O aplikaci	52
8.1	Měření přes noc	54
8.2	Přehled měření pana vedoucího ze září	55
8.3	Přehled mých vlastní měření z října	55
C.1	Sekvenční diagram připojení měřícího zařízení k mobilnímu přes Bluetooth	70

Seznam tabulek

3.1	Zkratky elektrodermálních měření [2]	16
-----	--	----

Kapitola 1

Úvod

Velkou součástí každodenního života člověka jsou emoce a reakce na různé podněty, mezi které patří škála různých konceptů, které se nacházejí v našich běžných životech jako například smrt či narození. Tyto koncepty dokáže člověk vnímat díky přirozeně vyvinutým smyslům, kterými jsou zrak, sluch, hmat, chuť a čich. Reakce a emoce jsou vytvořeny psychikou člověka a mohou ovlivnit lidské chování. Psychika člověka se nedá jednoduše studovat, avšak biologie člověka poskytuje mechanismus, který umožňuje zachycení a studování těchto vlastností. Tento mechanismus poskytují ekrinní potní žlázy, které jsou zabudovány do lidské kůže. Inervace těchto žláz napomáhá k změně elektrický vlastností těla. Tyto změny jsou známy jako elektrodermální aktivita či galvanická kožní reakce. Mezi elektrické vlastnosti získané z kůže patří vodivost, kterou lze změřit pomocí elektrod [1, 2].

V této práci se elektrody vyskytují na zařízení Sensatio [3], které dokáže zachytit a zasílat, pomocí technologie Bluetooth, informace o elektrodermální aktivitě. Zařízení Sensatio slouží pouze k zaslání těchto informací a tak je nutno využít technologii Bluetooth pro zaslání naměřených dat na výkonnější zařízení, jako například počítač či mobil. Získané data lze analyzovat a vizualizovat, což poskytne rozhraní pro biofeedback, tudíž zpětnou vazbu po měření elektrodermální aktivity, která může prohloubit znalosti měřícího o jeho psychickém stavu. Díky této zpětné vazby lze podniknout opatření k redukování například denního stresu, krevního tlaku [4] či dokonce epileptických záchvatů [5].

K poskytnutí rozhraní měření a biofeedbacku vzniká aplikace na mobilním zařízení. Důvodem vzniku aplikace pro mobilní zařízení je jeho vlastnost jednoduché přenositelnosti. Navíc se v dnešní době považují mobilní zařízení za nutnost. Je to silná a jednoduchá forma komunikace, zabalená v malém, přenosném zařízení. Své vlastní mobilní zařízení má více jak miliarda lidí a čísla stále stoupají. S tím přicházejí i nové technologie a platformy, zjednodušující denní používání těchto zařízení. Jednou z těchto platform je *Android*, která je vyvíjena jednou z největších společností na světě *Google*. *Android* je nejvíce rozšířená platforma mobilních zařízení. Její součástí není pouze operační systém, ale i nástroje k vývoji aplikací na tuto platformu. Tato práce bude implementovat aplikaci právě na této platformě.

V první části práce se vyskytuje popis pokožky a pocení, jejíchž funkčnost je nutno znát pro zkoumání elektrodermální aktivity. Obsahem druhá části práce je popis elektrodermální aktivity, její terminologie, typů měření a jejího využití. V třetí části práce se vyskytuje popis platformy Android, jak vznikla, jakou má architekturu a poté seznámení s jejími komponenty a vlastnostmi, které je nutné znát pro vývoj a návrh aplikace. V čtvrté části práce se vyskytuje popis jednotlivých technologií, vývojových prostředí a nástrojů, které budou použity během implementace aplikace. Součástí páté kapitoly je návrh aplikace, která opodstatňuje a analyzuje požadované i nepožadované funkce aplikace. Do návrhu je zahrnut i výběr vývojového prostředí a jazyka. Šestá část se zabývá implementací navržených funkcí. Její součástí jsou obrázky vzhledu aplikace, některé zdrojové kódy a vysvětlení práce jednotlivých funkcí. Předposlední, sedmá část se zabývá testováním aplikace. Testování je pro vývoj velmi důležité a je vhodné aplikaci vyzkoušet na co nejvíce zařízeních. Součástí předposlední části je také experimentální část, která vychází a popisuje měření z důsledku testování. V poslední části je shrnuta aplikace a výsledky této práce. Součástí poslední části je také popis možných budoucích rozšíření aplikace.

Kapitola 2

Kůže a pocení

Ještě před popisem měření vodivosti pokožky, čímž se tato aplikace zabývá, je nutné si popsat funkci kůže a pocení. Kůže, jakožto největší orgán v těle, společně s pocením jsou komplexními prvky lidského těla a v této práci budou zmíněny pouze ty funkce, jež jsou potřebné k měření.

2.1 Anatomie kůže

Lidská kůže je selektivní bariéra, která ochraňuje lidské tělo od různých infekčních agentů jako jsou viry či bakterie. Zároveň se využívá její selektivní schopnost propouštět nechtěné objekty ven z krevního oběhu. Pomáhá také regulovat teplotu lidského těla právě pomocí pocení.

Kůže samotná se skládá z několika komplexních částí. Celkově jsou tyto části tři a můžeme je vidět na obrázku 2.1. Tyto části jsou:

- Epidermis (pokožka)
- Dermis (škára)
- Hypodermis (Podkožní tuk, podkožní vazivo)

2.1.1 Epidermis

Tato vrstva, tlustá přibližně 0,1 mm, slouží jako ochrana před ztrátou vody, toxiny a bakteriemi. Je také neustále měnícím se prostředím, kde se buňky oddělené od spodní (bazální vrstvy) posouvají směrem nahoru a přitom procházejí řadou změn. Tento proces se nazývá keratinizace (rohovění) a rozkládá pokožku na několik dalších částí:

- Stratum corneum (bazální vrstva)
- Stratum lucidum (vrstva ostnitých buněk)
- Stratum granulosum (vrstva zrnitých buněk)

- Stratum spinosum (vrstva jasných buněk)
- Stratum basale (rohová vrstva)

Kvůli absenci krevních cév nemůžou tyto vrstvy získat živiny z krve, avšak můžou je získat z nižší tkáně pomocí difuze. Buňky postupem času spotřebují všechny živiny, umřou a odštěpí se od kůže. [2, 6, 7, 8]

2.1.2 Dermis

Dermis, neboli Škára, dosahuje tloušťky až 2,5 mm. Oproti pokožky je bohatě prokrvená, tudíž buňky mají dostatek živin pro svou funkci a mohou nasycovat epidermis. Dělí se na dvě části:

- Stratum reticulare - spodní vrstva, volná hranice s podkožím
- Stratum papillare - vrchní vrstva, vlnitá hranice s epidermis

Její součástí jsou krevní cévy, nervová zakončení, žlázy a vlasové folikuly. Tato vrstva je díky složek jako kolagen a elastin silná a pružná. Jejími hlavními funkcemi jsou termoregulace, citění díky nervovým zakončením a díky své pružné struktuře také ochrana těla. [9]

2.1.3 Hypodermis

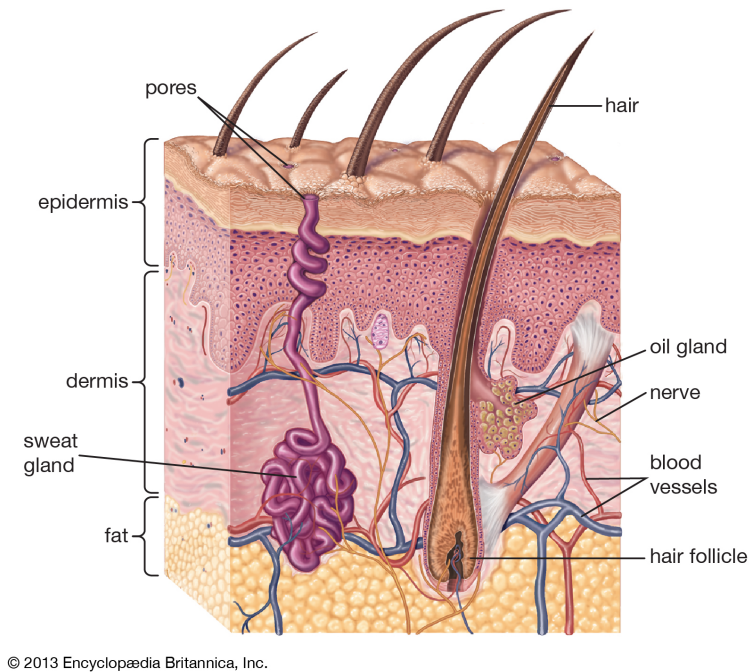
Hypodermis, neboli podkoží či podkožní vazivo, je nejspodnější vrstva kůže, která díky své struktuře a složení z tukových buněk a speciálních kolagenních vláken, pomáhá tělu uchovávat energii a chránit ho díky vztlaku například před nárazy [2, 7].

2.2 Pocení

Funkce kůže slouží primárně pro termoregulaci těla savců či indikace emocionálního vzrušení. Provádí se sekrecí vody z míst po celém těle díky dvou druhů potních žláz, tedy apokrinních a ekrinních, zabudovaných do kůže. Termoregulace se aktivuje v případě, že tělo dosáhne 28-30 °C a její funkce zesiluje se zvyšující se teplotou lidského těla.

2.2.1 Apokrinní žlázy

Žlázy, které se nacházejí v okolí prsou, podpaží a genitálií. Jejich sekrece je z těla vyváděna přes vlasové folikuly a celkově jsou méně četné a činné z pohledu pocení než ekrinní. Apokrinní žlázy nehrají moc významnou roli v rámci elektrodermálních měření, jelikož jejich sekrece není aktivována nervovým systémem ale adrenalinem [2, 11]. V rámci zvířat je jejich aktivita spojována s produkcí sexuálních hormonů (feromonů), avšak jejich činnost pro měření emocionálního vzrušení v lidech je stále diskutována [1].



Obrázek 2.1: Struktura kůže [10]

2.2.2 Ekrinní žlázy

Ekrinní žlázy jsou exokrinní, tudíž jejich sekrece (pot) je vylučována přímo na povrch pokožky. V lidském těle jsou v počtech od dvou až do pěti milionů s průměrnou distribucí 150-340 na centimetr čtvereční. Jejich největší četnost je ovšem na čele, dlaních a chodidlech. Ekrinní žlázy procházejí lidskou kůží, příklad lze vidět na obrázku 2.1 (Sweat gland). Sekreční část se nachází v hypodermu, následována trubicí, procházející přes dermis, a spirálovitě stočeného vývodu v epidermu [8].

Funkce ekrinních žláz je primárně určena pro termoregulaci, avšak na místech jako jsou chodidla a dlaně je spíše spojována s psychickými podněty, díky jejich velké koncentraci na těchto místech. Jejich inervace je způsobena sympatickou částí autonomního nervového systému (ANS) [1, 2].

Kapitola 3

Principy elektrodermálních jevů

Elektrodermální aktivita (EDA), také známá jako galvanická kožní reakce (GSR), je nejčastěji využívaná metoda pro měření změn elektrických vlastností kůže, způsobeny psychickými podněty jako je například strach.

3.1 Historie elektrodermální aktivity

Zájem o studii elektrodermálních jevů začal už před počátkem 19. století, kde vědci jako Féré a Tarkhanov demonstrovali změny v elektrické rezistenci kůže, které byly způsobeny smyslovými a emocionálními podněty [12]. Férého zjištění o poklesech odporu kůže díky vizuálním, sluchovým a čichovým podnětům, vzniklo po připojení dvou elektrod, umístěných na povrchu kůže, a puštěním malého elektrického proudu mezi nimi. Zjistil, že se kůže stává lepším vodičem elektřiny při představení vnějších podnětů. Tarchanoff krátce poté zjistil, že se elektrický potenciál kůže mění i bez aplikace tohoto proudu [1].

Postupem doby a s příchodem nových a lepších měřících zařízení, se GSR spojovalo čím dál tím více s psychickou aktivitou. V 60. letech bylo GSR již přijímáno jako standard pro měření emocionálního vzrušení či stresu. Následovalo období konsolidace, kdy příslušné mechaniky, společně s tepovou frekvencí a krevním tlakem, byly objasněny pro další generaci měření psychických podnětů [12].

3.2 Terminologie

EDA byla poprvé představeno Johnsonem a Lubinem v roce 1966 jako společný termín pro všechny elektrické jevy kůže, avšak vznikla potřeba pro rozdělení různých typů měření po objevech Férého a Tarkhanova, tudíž o rok později byl promítnut návrh na standardizaci, která je v dnešní době akceptována jako aktuální. Tato standardizace se týkala rozdělení na měření bez proudu (endosomatic) a na měření se střídavým či stejnosměrným proudem (exosomatic) [2].

	Endosomatic	Exosomatic			
	Potenciál	Stejnoseměrný proud		Střídavý proud	
		Odpor	Vodivost	Impedance	Admittance
Obecně	SP	SR	SC	SZ	SY
Tonic (úroveň)	SPL	SRL	SCL	SZL	SYL
Phasic (odezva)	SPR	SRR	SCR	SZR	SYR

Tabulka 3.1: Zkratky elektrodermálních měření [2]

Na vzniklou terminologii se můžeme podívat v tabulce 3.1. Zkratky začínají písmenem S (Skin) a poté následuje co se měří, tudíž R (Resistance), C (Vodivost), Z (Impedance) a Y (Admittance). Poslední část je určena podle délky odezvy. P (Phasic) určuje krátkodobé změny v elektrodermální aktivitě, kdežto T (Tonic) určuje dlouhodobé absolutní změny po celou dobu stimulace. Zkratky se poté ještě liší v závislosti, zda je měření nespecifické (NS) či související s událostí (ER). Společně s NS a ER byly z tabulky vynechány i doplňkové zkratky, které určují frekvenci, amplitudu, zpoždění, dobu zotavení a dobu náběhu [12].

3.2.1 Exosomatické měření se stejnosměrným proudem

Měření se stejnosměrným proudem je nejvíce používaná a preferovaná metoda v oblasti měření elektrodermální aktivity ať už se jedná o měření tonic (úroveň) či phasic (odezva) komponent [13].

3.2.1.1 Kožní odpor (SR) a vodivost (SC)

Pro měření je nutno použít dvě elektrody, většinou ze stříbra/chloridu stříbrného společně s elektrolyty (sodík/draslík). Poloha těchto dvou elektrod je většinou na jednom místě vedle sebe, nejčastěji na rukách, chodidlech a čelu [13].

Výsledky měření jsou ve většině publikací převedeny na jednotky vodivosti, z důvodu lehčího zpracování statistik [2]. Tyto výsledky je možné převést díky Ohmova zákona (3.1), kde V je napětí, I je proud a R je odpor. Díky konstantní hodnotě aplikovaného napětí se za napětí dá dosadit číslo 1, díky čemuž je proud ve vztahu nepřímé úměrnosti s odporem [13].

$$V = IR \quad (3.1)$$

Jednotka vodivosti je SI (siemens, kdysi mho) a jednotka odporu je Ω (ohm). Při měření se většinou používá malé napětí (0.5 V) [13]. Hodnoty SCL se pohybují kolem 5-20 $\mu\text{S}/\text{cm}^2$. Hodnoty SCR jsou závislé na oblasti, kde se používá elektroda. V těchto místech je jejich hodnota něco mezi .001 μS až 5 $\mu\text{S}/\text{cm}^2$. Zpoždění se pak pohybuje mezi 1.3 až 2.5 s [11]. Hodnoty odporu v kůži se pohybují ve vysokých hodnotách (stovky k Ω). Při měření se většinou používá malý rezistor od 200 až do 1000 Ω [13].

3.2.2 Exosomatické měření se střídavým proudem

Měření se střídavým proudem může obejít problémy polarizace, které se vyskytují u měření se stejnosměrným proudem. Ačkoliv je tento problém řešen nepolarizujícími elektrodami (Ag/AgCl), tak existují obavy, že toto řešení může jen částečně zabránit polarizaci a tím mohou vzniknout problémy. Další výhodou je současné měření SP a SC na jediné elektrodě [14].

Měří se zásadně dvě veličiny, odporová část střídavého proudu, označena jako impedance (Z), měřena v jednotkách kiloohmů a vodivostní část střídavého proudu admitance (Y) měřena v jednotkách mikrosiemen [13].

Ačkoliv přechod na toto měření může být výhodné, v současné době není dostatek studií, které se jim zabývají a k jejímu porozumění je potřeba větší matematické odborné způsobilosti než u měření se stejnosměrným proudem [13].

3.2.3 Endosomatické měření

Měření bez externího proudového zdroje jsou nazývána endosomatické. Touto technikou se měří pouze potencionální změny v kůži samotné [2]. Jako výhodu můžeme poznačovat právě nepotřebnost speciálních zesilovacích systémů a podle [13] mohou být provedeny i elektroencefalografií (EEG). Výsledné SPR jsou však komplikované a těžké na interpretaci.

3.2.3.1 Kožní potenciál (SP)

Pro měření kožního potenciálu je nutno použít dvě elektrody s elektrolyty, které jsou složeny ze stejných materiálů, jako dříve zmíněné elektrody pro měření vodivosti kůže. Tyto elektrody jsou připojeny na zesilovač napětí [12]. Pozice elektrod je oproti měření vodivosti trochu jiná, jedna elektroda by měla být na aktivním místě měření (většinou dlaň) a druhá na neaktivním místě pro referenci.

Hodnoty SPL se většinou pohybují kolem 10 až -70 mV [11]. Hodnoty SPR mohou být negativní (větší negativní potenciál na aktivní pozici - ruka) i pozitivní. SPR mohou být jednofázové i dvoufázové [13].

3.2.4 Využití EDA

Změna elektrických vlastností kůže může v našem těle znamenat mnoho věcí, postupem času vzniklo několik studií, zabývajících se reakcemi na různé podněty, jako například emoce, pracovní zátěž či stres [12]. Výsledky EDA měření, společně s hodnotami krevního tlaku či napětí svalů, pak mohou být brány jako biofeedback, tudíž získání zpětné vazby po měření elektrodermální aktivity, díky čehož lze podniknout kroky k redukování například denního stresu, krevního tlaku [4] či počtu epileptických záchvatů [5].

Kapitola 4

Android

Android je mobilní operační systém, založen na linuxovém jádře (Linux kernel). Android je open-source, tedy zdrojový kód je vydán pod licencí, která umožňuje uživatelům přistoupit k tomuto zdrojovému kódu zdarma a upravovat ho podle svých potřeb.

4.1 Historie

Android byl založen v říjnu roku 2003 v Kalifornii. Jeho zakladateli byli Rich Miner, Nick Sears, Chris White a Andy Rubin. Android Inc byl v srpnu roku 2005 odkoupen společností Google a to včetně jejich zaměstnanců. Pod vedením Andyho Rubina bylo v roce 2007 vytvořeno první mobilní zařízení platformy Android, běžící na linuxovém jádře. V listopadu roku 2007, po představení platformy Android, bylo také vytvořeno sdružení Open Handset Alliance, jehož součástí byli HTC, Motorola a Samsung. Cílem tohoto sdružení bylo vyvíjení rozsáhlých open-source platform [15].

4.1.1 Verze

Po dobu vývoje platformy Android vzniklo mnoho verzí [15], které přinesly funkcionality jako připojení mobilu pomocí USB k počítači, Near field communication (NFC), více kamer a jiné. Detailní popis každé verze jde nad rámec práce a v této části budou popsány pouze ty verze, které je nutno znát pro implementaci aplikace.

Verze 4.3 (Jelly Bean) je verze, která sebou přinesla podporu pro Bluetooth s nízkou spotřebou energie (BLE), z anglického Bluetooth low energy. Součástí této podpory je posílání informací a nacházení zařízení, služeb a charakteristik.

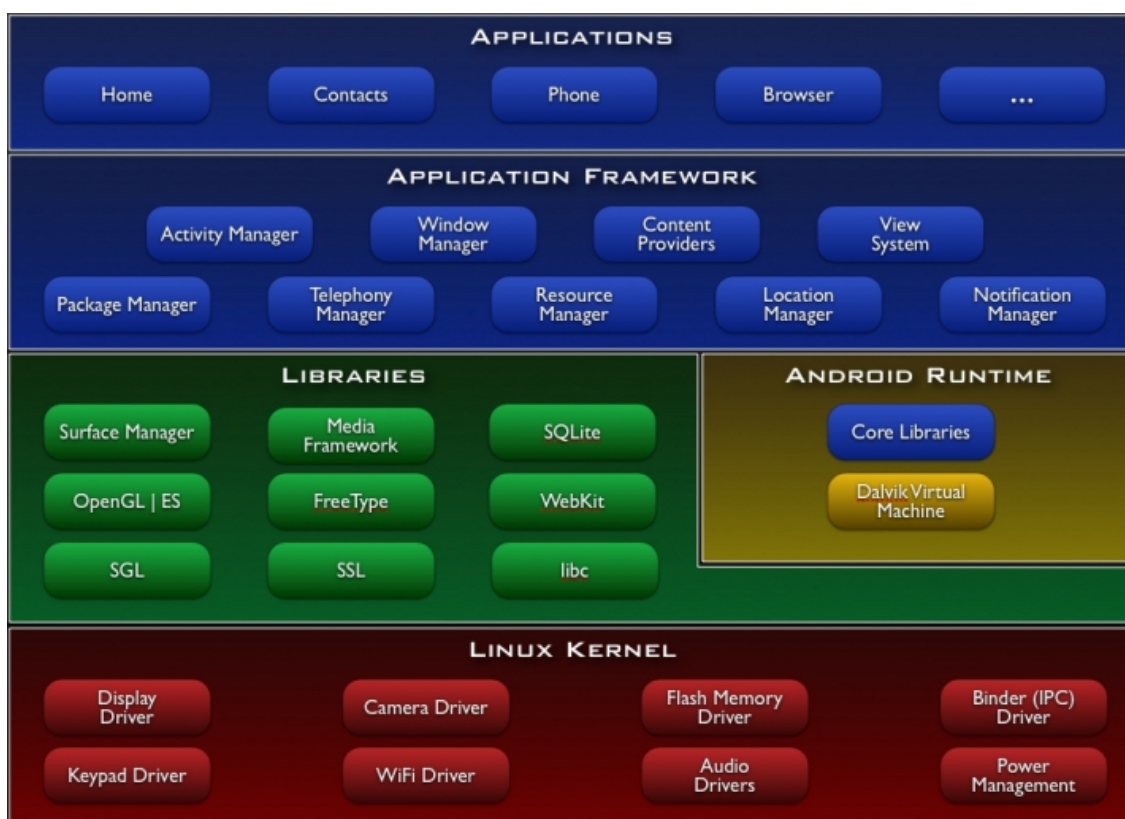
Verze 6.0 (Marshmallow) je verze, která mění způsob získávání oprávnění pro aplikaci. Mobily s verzí nižší než tato udělovaly všechna oprávnění během instalace aplikace. Mobily s touto verzí a vyšší musí udělit některá oprávnění až během běhu aplikace.

Verze 8.0 (Oreo) přivedla změnu notifikací v mobilech. Aby se notifikace vůbec ukázaly v této verzi Androidu, tak je nutno vytvořit notifikační kanál.

Největší podíl na trhu mají podle [16] v současnosti mobily s verzi Androidu 10.0, jejich celkový podíl se blíží k 36%. Dalších 35% se dělí mezi verze 8.0 (Oreo) a 9.0 (Pie).

4.2 Architektura

Základní architektura Androidu se skládá z pěti vrstev [17] a lze jí vidět na obrázku 4.1. Popis jednotlivých částí se nachází níže.



Obrázek 4.1: Architektura operačního systému Android [18]

4.2.1 Linux kernel

Linux kernel je jádro operačního systému Android. Jeho hlavní součástí je abstrakce mezi použitým hardwarem a softwarem, tudíž ovladače zařízení, řízení paměti a procesů a jiné.

4.2.2 Knihovny

Tato vrstva obsahuje knihovny, které poskytují základní funkcionalitu pro vývoj aplikací. V následujícím seznamu se pokusím zmínit některé knihovny, které poskytují základní funkčnost vyvíjené aplikace:

android.os - knihovny poskytující informace o verzi, funkce pro předávání zpráv a jiné.

android.content - knihovny obsahující celo-systémovou, či aplikační komunikaci

android.bluetooth - knihovny umožňující využití práci s Bluetooth jako vyhledávání zařízení a připojení se k nim

android.widget - knihovny obsahující základní prvky uživatelského rozhraní (UI)

android.media - knihovny poskytující práci se zvuky a videi

androidx.lifecycle - novější JetPack knihovny, které pomáhají ke konzistentnímu kódu napříč aplikacemi. Lifecycle knihovny si jsou vědomy životního cyklu aplikace a pomáhají její správě.

4.2.3 Android runtime

Android runtime obsahuje virtuální stroj Dalvik (DVM), anglicky Dalvik virtual machine, který je důležitý pro překlad zdrojového kódu jazyka Java, ve kterém jsou aplikace vyvíjeny, do bytekódu, který pak využívají aplikace k jejich spuštění a práci. Dalvik používá just-in-time (JIT) kompilaci, která dynamicky překládá Dalvik bytekód do kódu strojového.

V novějších verzích (5.0+) je součástí systému i Android runtime (ART), který přináší nové funkce ladění, lepší garbage collection a jiné. ART používá novější a výhodnější ahead-of-time (AOT) kompilaci, která překládá bytekód pouze jednou během instalace, čímž zlepšuje běh aplikace.

Android runtime pak ještě obsahuje základní knihovny jazyka Java.

4.2.4 Application Framework

Nabízí velkou škálu služeb, které vývojáři mohou používat ve svých aplikacích. Jeho součástí je například View systém, který umožňuje tvorbu prvků uživatelského rozhraní jako jsou tlačítka, seznamy a jiné.

4.2.5 Vrstva aplikace

Poslední vrstva, která obsahuje jednotlivé aplikace, které si uživatel může spustit a používat.

4.3 Komponenty

Základem programování v Androidu jsou komponenty. Komponenta může být cokoliv od tříd pro interakci s uživatelem (aktivity) až po komunikaci mezi komponenty (intenty). V této části se pokusím shrnout základní komponenty, které jsou v práci využity.

Activity je hlavní třída pro interakci s uživatelem. Je implementována pomocí třídy *Activity*.

Fragment je nezávislá třída, běžící v kontextu aktivity. Jejím cílem je opakované použití v aktivitách.

Service je třída navržena pro práci v pozadí. Není závislá na životním cyklu aplikace a implementuje se pomocí třídy *Service*.

Intent je pasivní datová struktura, která drží abstraktní popis operace, která by se měla spustit. Slouží pro celosystémovou (explicitní intenty) či aplikační komunikaci (implicitní intenty).

Broadcast receiver je subscriber ze vzoru publish–subscribe, tudíž umí přijímat události, které byly vyslány jednomu či více příjemcům asynchronně. Aby aktivita či služba mohla začít přijímat data, je nutné ho registrovat a po použití také odregistrovat.

Notification je třída pro ukazování notifikací na stavovém řádku mobilu. Vyžadují jí některé jiné komponenty, jako například služba.

Messenger je reference na Handler, který můžou využít ostatní komponenty pro komunikaci.

Bundle je třída, která funguje jako balíček s Parcelable daty. Většinou se využívá pro posílání dat mezi aktivitami za pomoci intentů.

Toast malé okno vyskakující na spodní části obrazovky. Slouží převážně pro zpětnou vazbu.

4.4 Vlastnosti OS Android

Tato část bude povídat o základních vlastnostech OS Android jako například životní cyklus či priorita aplikací.

4.4.1 Životní cyklus aktivit

Velkou součástí Androidu jsou aktivity, které mají svůj vlastní životní cyklus [19]. Tím je myšleno, že pokud je aktivita spuštěna, dána na pozadí či vyhozena z paměti, tak prochází funkcemi, které slouží k ošetření daných situací. Tyto funkce a jejich pořadí, ve kterém se vyvolávají, lze vidět na ilustračním obrázku 4.2.

Důležité je vědět, kdy nastane která situace a kdy se zavolá jaká metoda. Například jednou z nejdůležitějších metod je metoda *onCreate*, která se volá během tvorby aktivity. V této metodě se většinou inicializují všechny komponenty aktivity. Další důležitou metodou je metoda *onPause*, která se volá, když uživatel naviguje do jiné aktivity či jiné aplikace a metoda *onResume*, která se volá když naviguje zpátky do pozastavené aktivity. Jako poslední stojí za zmínku metoda *onDestroy*, ve které se většinou odstraňují zdroje a reference aktivity.

Během změny orientace mobilu projde aktivita celým životním cyklem, tudíž je zničena a poté znovu vytvořena. Jelikož chceme některá data zachovat i přes toto zničení, tak je možno využít komponenty OS Android, které si jsou vědomy životního cyklu aktivit:

onSaveInstanceState je nejjednodušším řešení pro uchování dat po ukončení životního cyklu aktivity. Ve verzích vyšších než 9.0 se vyvolává po metodě *onStop* a v nižších verzích před *onStop*.

onRestoreInstanceState je metoda, která se vyvolává po metodě *onStart*. Dají se zní získat data, které byly uloženy pomocí *onSaveInstanceState*.

Fyzické úložiště se dá použít díky zabudovaným třídám jako *SQLite* (databáze), *SharedPreferences* (dvojice klíč-hodnota) a metody a třídy z knihovny *java.io*. Všechny tyto třídy se používají, avšak jinými způsoby, k uložení dat do vnitřní paměti mobilu.

ViewModel je třída, která se pojí s aktivitami za účelem uložení a správy dat. Je také vhodná pro propojení a komunikaci aplikační vrstvy s ostatními vrstvami třívrstvé architektury.

4.4.2 Životní cyklus aplikace

Každá část aplikace má podle platformy Android [21] danou šanci na ukončení svého životního cyklu [21]. Systém může zastavením aplikace získat zpátky své zdroje (paměť) a tím je využit v jiných aplikacích. Priorita, pomocí které se vybírají aplikace, které budou pracovat dále, je dána takto (seřazeno od největší priority po nejmenší):

- Proces v popředí - proces je v popředí pokud uživatel interaguje s aktivitou, přijímá se celosystémový či aplikační přenos pomocí metody *BroadcastReceiver.onReceive*, nebo když probíhají základní metody služeb (*onStart*, *onDestroy*, *onCreate*).
- Viditelný proces - viditelný proces je pozastavená aktivita (například když je dialog v popředí) či služba v popředí (posunutý do popředí pomocí metody *Service.startForeground*).
- Proces služby - služba, která není v popředí (spuštěna pomocí metody *startService*).
- Procesy v mezipaměti - nepotřebné procesy, například aktivity, které nejsou viditelné (byla zavolána metoda *onStop*).

4.4.3 Služby

Jak už bylo dříve řečeno, služba je komponenta aplikace, vhodná pro dlouho trvající operace, běžící na pozadí (může běžet i přes přepnutí na jiné aplikace). Služby mají několik stavů, které mění jejich životní cyklus a celkové chování:

- Služba vázaná - služba se může navázat na jednotlivé aktivity a komunikovat s nimi.
- Služba spuštěná - dělí se na dříve zmíněné služby v pozadí a v popředí.

Služba se vytvoří po jejím spuštění či navázání na aktivitu. Služba zaniká pouze v případě, že byla zastavena příkazem *stopService* či *stopSelf* a není vázána s žádnou aktivitou.

4.4.4 Struktura projektů

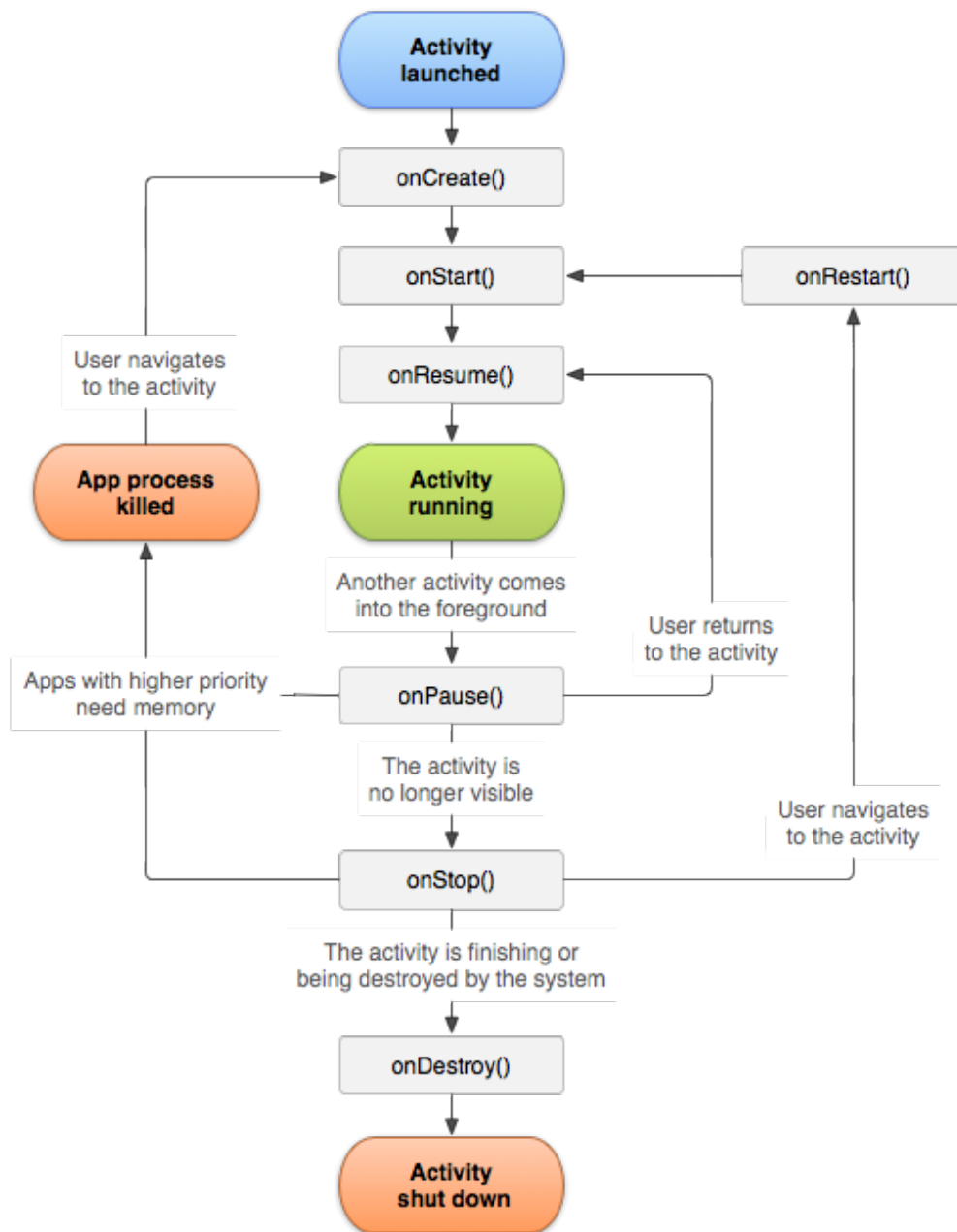
Samotné projekty mají strukturu danou platformou Android [22]. Projekty se dělí na složky, které obsahují kódy, obrázky a XML obsahující UI.

Jednou z nejdůležitějších částí projektu, bez které by aplikace nefungovala, je soubor *AndroidManifest.xml*, který popisuje základní informace o aplikaci jako jsou komponenty služeb a aktivit společně s referencemi na související třídy implementující tyto komponenty, ikona a název aplikace a všechny druhy oprávnění.

Určení mobilní verze, kterou bude aplikace používat, pluginy a všechny použité balíčky (závislosti) se dají nalézt v souboru *build.gradle*, který se nachází v kořenovém adresáři.

Samotný kód se píše do souborů, odpovídající jazyku aplikace (například .java v případě Javy), ve složce *src/XY*, kde XY označuje ID aplikace, která taky mimo jiné určuje názvosloví hlavní složky ve fyzickém úložišti mobilu.

XML obsahující popis UI, obrázky a jiné grafické materiály, řetězce, pole řetězců a čísel, položky menu a další se nacházejí ve složce *res*.



Obrázek 4.2: Ilustrace životního cyklu aktivit [20]

Kapitola 5

Použité technologie, vývojové nástroje a prostředí

V této části bych chtěl popsat nástroje a vývojové prostředí, které se používají k tvoření aplikací pro platformu Android. Dále bych chtěl popsat měřicí zařízení Sensatio, které se v aplikaci používá k měření GSR a technologii BLE (Bluetooth s nízkou spotřebou energie) [23], z anglického Bluetooth low energy, která se používá pro komunikaci mezi zařízeními. Jako poslední je nutno zmínit souborový formát CSV, kterou aplikace používá pro ukládání dat, statistik a analýz.

5.1 Android studio

Android studio [24] je vývojové prostředí specificky vytvořeno pro vývoj aplikací na OS android. Je vybudováno na základě IntelliJ IDEA a je dostupné ke stažení na platformách jako jsou Windows, MacOS a Linux.

Pro podporu vývoje využívá sestavovací systém Gradle, podepisování aplikací, layout editor, AVD a mnoho dalších. Mezi známější jazyky, používaných pro tvorbu android aplikací, se řadí Kotlin, Java a C++.

5.2 JDK

Sada pro vývoj Java aplikací (JDK), z anglického Java development kit, [25] je soubor nástrojů a knihoven sloužících pro tvoření a spouštění Java programů. Jeho součástí jsou Java virtual machine (JVM) a Java runtime environment (JRE), obojí sloužící k spouštění Java programů. Dále pak jsou jeho součástí vývojové nástroje jako překladač a debugger.

5.3 Android software development kit (SDK)

Sada pro vývoj softwaru (SDK), anglický také známa jako Software development kit, je kolekce nástrojů, většinou v jednom balíčku, využívaných pro vývoj aplikací na různé druhy operačních systémů či hardwarových platform. Android SDK [26] tedy poskytuje možnost vývoje aplikací pro OS Android. Součástí tohoto balíčku jsou návody, dokumentace, ukázky kódů a nástroje jako debugger a kompilátor, sloužící pro ulehčení vývoje a testování aplikací.

5.4 AVD

Virtuální zařízení Androidu AVD, z anglického Android virtual device, [27] je nástroj pro emulaci opravdového mobilního zařízení s OS android na počítači. Dá se díky němu replikovat hardwarové a softwarové nastavení zařízení, běžících na různých verzích systému Android. Tento nástroj tedy zaručuje schopnost testování bez nutnosti vlastnit zařízení se systémem Android.

5.5 XML

Rozšiřitelný značkový jazyk (XML), z anglického Extensible markup language, je značkový jazyk, tudíž řada kódů či tagů, sloužících pro popis digitálního dokumentu. Za jeho standardizací stojí World Wide Web Consortium (W3C), kteří se tento jazyk snažili udělat univerzální, multiplatformní a jednoduchý na pochopení. Mezi jeho soupeře se dá považovat například JSON, avšak co se týče Android aplikací, dává se přednost právě XML hlavně při tvoření layoutů, nebo definování řetězců či hodnot.

5.6 CSV

CSV [28], z anglického Comma-separated values, je typ souboru, který odděluje záznamy řádky a samotné hodnoty čárkami. CSV soubory jsou ve formě prostého textu (plain text), díky čemuž se dají otevřít pomocí prakticky všech textových editorů, příkladem může být třeba Notepad či Excel. Znaků používaných pro psaní dat do CSV souboru mohou být zakódovány pomocí Unicode, například UTF-8, či ASCII standardu.

5.7 BLE

Bluetooth [23] je technologický standard určen pro bezdrátovou komunikaci mezi dvěma či více (Bluetooth multipoint) zařízeními. Tyto zařízení mohou být notebooky, počítače, tiskárny, sluchátka, klávesnice a další.

BLE je rozšíření technologie Bluetooth, které bylo součástí její 4.0 specifikace. Jejím cílem bylo snížit energetické nároky na přenos dat mezi zařízeními. Zařízení používající tento technologický standard by měla být sama o sobě méně energeticky náročná s očekávanou životností v řadách měsíců či roků. Nejčastější využití této technologie je v zařízeních, které potřebují být zapnuta delší dobu. Příkladem mohou být různá zdravotnická zařízení, senzory pohybu a mnoho dalších.

5.7.1 Základy funkce BLE

BLE se řídí pomocí protokolů, sloužících pro připojení zařízení a přenos dat. Pro propagaci a připojení zařízení se využívá Obecný profil přístupu (GAP) [29], anglicky Generic Access Profile, který definuje role připojených zařízení. Tyto role jsou:

- Periferní - malá zařízení jako například monitor srdečního tepu nebo právě GSR zařízení. Mají nízkou spotřebu energie a připojují k centrálnímu zařízení, kterému většinou posílají data.
- Centrální - zařízení s větší výkonem a pamětí jako například mobil.

Obě dvě tyto zařízení se dokážou propagovat, tudíž zajišťovat, že je ostatní zařízení budou moci vidět při skenování, avšak při BLE připojování propagaci většinou provádí pouze periferní zařízení. Skenování, které pomáhá najít propagované zařízení, většinou provádí centrální zařízení. V povaze je [30], že periferní zařízení může být připojeno pouze k jednomu centrálnímu zařízení.

Po připojení zařízení je možno provést operace jako čtení, psaní. Jakým způsobem lze tato data vyměnit určuje Obecný profil atributů (GATT) [30], anglicky Generic Attribute Profile, který využívá protokol atributů (ATT), anglicky Attribute Protocol, pro ukládání profilů, služeb a charakteristik. Z pohledu GATT lze tedy klasifikovat dvě připojené zařízení jako klient a server, kde server drží ATT data pro vyhledání pomocí vyhledávací tabulky a klient posílá příkazy nebo žádosti na server. Komunikace je v případě GATT dvousměrná, tudíž server může posílat data klientovi a naopak.

Samotné transakce obsahují objekty nazvanými profily, služby a charakteristiky. Popsat tyto objekty lze takto:

- Profily - abstrakce definovaná výrobcí periferních zařízení či Bluetooth SIG, která obsahuje jednu či více služeb.
- Služby - rozdělování charakteristik do logických celků. Příkladem může být GSR zařízení se službou obsahující informace o zařízení (výdrž baterie, výrobce), či službou obsahující naměřená data.
- Charakteristiky - pole dat

Služby i charakteristiky mají jedinečný a univerzální identifikátor (UUID), anglicky Universally unique identifier, díky kterému je lze vyhledat a rozeznat.

5.8 Zařízení Sensatio

Zařízení Sensatio [3], mezi které patří myš a náramek, slouží k měření vodivosti kůže a variability srdeční frekvence. V této práci se bude používat pouze náramek, který lze vidět na obrázku 5.1. Náramek se skládá z BLE senzoru a dvou kontaktů (elektrod).

Náramek se spouští po nasazení jeho dvou elektrod na prsty. Po navázání připojení s jiným zařízením z něho lze číst služby a charakteristiky. Náramek se sám odpojí v případě, že elektrody nejsou nasazené na prstech. Doba, po které se odpojí, se pohybuje kolem dvou minut.

Náramek také potřebuje chvíli na to, aby normalizoval své hodnoty po nasazení na prsty. Aby měření probíhalo s normalizovanými hodnotami, tak je vhodné počkat přibližně 60 sekund.

Data ze zařízení lze číst opakovaně. Opakované čtení ze zařízení je nutno limitovat, aby se stihlo aktualizovat jak mobilní tak měřící zařízení. Čas, ve kterém lze bezztrátově získat nové data, je přibližně 100 milisekund.



Obrázek 5.1: Náramek Sensatio [3]

Kapitola 6

Návrh

Tato kapitola se zabývá návrhem řešení základních problémů, které bylo nutno vyřešit ještě před implementací aplikace. Samotný návrh se točí kolem zadání této práce, jelikož je jeho obsahem nutná funkčnost aplikace, a požadavky koncových uživatelů. Koncoví uživatelé jsou v tomto případě ti, kteří vlastní nebo mají zapůjčeno zařízení Sensatio, blíže popsaného v sekci 5.8.

6.1 Účel práce

Pro zařízení Sensatio existuje aplikace na počítač, která se používá pro měření GSR za účelem biofeedbacku či reakcí na různé psychické podněty. Měření pomocí počítače je vhodné pro hromadné či domácí měření, avšak jeho nevýhodou je nepřenositelnost měřícího zařízení (počítače). Díky přenositelnosti, snadnosti použití náramku Sensatio a potřebě získávat data na každodenní bázi, je vhodné vytvořit aplikaci na mobilních zařízeních, která ulehčí měření. Během návrhu byla ve vývoji aplikace na mobilní zařízení se systémem iOS, tudíž bylo nutno zajistit měření i na zařízeních se systémem Android.

6.2 Požadované funkce aplikace

Jak už bylo dříve zmíněno, cílem aplikace je poskytnout přenositelnou a jednoduchou možnost měření, proto je nutno zajistit následující funkce:

- Aplikace by se měla umět napojit na měřící zařízení Sensatio.
- Aplikace by měla umět získávat a ukládat naměřená data z měřícího zařízení Sensatio.
- Naměřená data by měla být uložena v CSV souborech, které by měly být přístupné i mimo aplikaci.

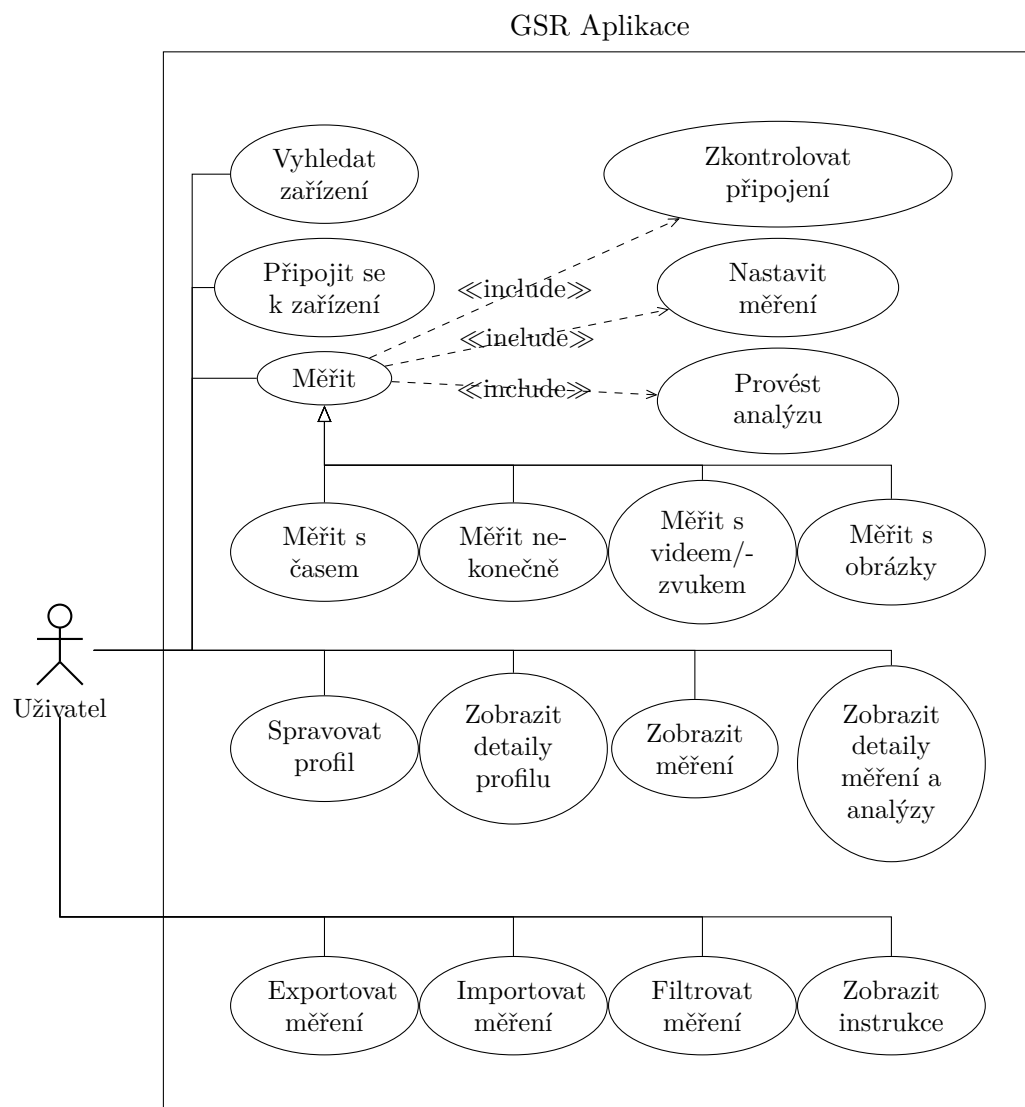
- V případě zájmu uživatele, použít jedno mobilní zařízení pro měření více osob, bude také poskytnuta možnost vytvoření profilů, která umožní rozdělení jednotlivých měření podle toho, kdo zrovna měřil.
- Biofeedback a reakce lze zajistit během měření, tudíž aplikace musí nabízet více druhů měření (s videem, zvuky, obrázky, časované bez jakýchkoliv podnětů).
- Pro měření by mělo jít nastavit jak často se bude číst ze zařízení Sensatio, jaká bude doba normalizace, jaké měl uživatel pocity před měřením a jiné.
- Aplikace by měla umožnit základní analýzu dat, složitější analýzy budou vypočítány pomocí knihovny, poskytnuté Ostravskou univerzitou. Pro uživatele je taky vytvořena příručka pro práci s aplikací.
- Poskytnutí možností přenášení (exportu) analýz a naměřených dat, za účelem další práce s těmito daty.
- Jednotlivá měření a analýzy by měly jít zpětně zobrazit.
- Uživatelům by měla být poskytnuta příručka pro práci s aplikací.

6.3 Dodatečné funkce aplikace

Mezi dodatečné funkce systému patří následující funkce:

- Rychlé připojení na zařízení Sensatio.
- Možnost nastavení aplikace.
- Zobrazení vytvořené příručky v aplikaci.
- Importování měření.
- Poskytnutí zpětné vazby ve formě zvuků, popisů či upozornění.
- Nekonečné, nebo taky neomezené měření.
- Podpora dvou jazyků (angličtina a čeština).
- Podpora dvou orientací mobilu.

V schématu 6.1 jsou zobrazené jednotlivé funkcionality aplikace.



Obrázek 6.1: Use case aplikace

6.4 Výběr programovacího jazyka a prostředí

Jak už je ze zadání známo, cílová platforma je Android. Pro vývoj aplikací na tuto platformu se dá využít několika vývojových prostředí, příkladem jsou právě Eclipse a Android studio. Pro implementaci jsem se rozhodl využít Android studio.

Co se týče programovacího jazyka, tak se opět naskytuje více možností jako například Kotlin či Java. Pro vývoj jsem si však vybral Javu.

6.5 Jazyk aplikace

Aplikaci budou převážně využívat občané České republiky, tudíž je vhodné pro tyto uživatele poskytnout aplikaci v češtině. V případě využití aplikace jinými zeměmi či podle chůce uživatele by v aplikaci měla být angličtina.

6.6 Verze androidu

Hlavním důvodem proč se zabývat verzí systému je technologie BLE (viz. 5.7). Google [31] dodal podporu pro BLE až ve verzi OS Android 4.3 (Jelly Bean), tudíž to aspoň pro začátek určuje minimální verzi systému, pro kterou bude tato aplikace vytvářena a můžeme kolem této informace orientovat implementaci.

6.7 Orientace mobilu

Některé aplikace mají pouze jednu orientaci, tedy na šířku nebo na výšku, avšak možnost otočit aplikaci byla z mého pohledu cenná, tudíž jsem se rozhodl ji implementovat.

Mnou vybrané řešení, které by zároveň zlepšilo architekturu systému tím, že by poskytlo oddělení obav (SoC), anglicky také známo jako Separation of concerns, jsou právě Android *ViewModely*. *ViewModely* krátce řečeno umožňují uchovávat data s vědomím životního cyklu androidu, tudíž dokážou uchovat data i přes zničení aktivit. Mým návrhem tedy bylo rozdělit aplikaci do částí, tudíž UI části aplikace by držely pouze logiku UI a interakce s aplikací, kdežto *ViewModel* by se staral o logiku aplikace a zaručoval by komunikaci s modely a daty (*FileSystem* soubory).

6.8 Profily v aplikaci

Jelikož je navrženo, že na jednom mobilu může měřit více uživatelů, tak je nutno zajistit profilovou hierarchii. Jednotlivé profily by se měly dát vytvořit, smazat a zobrazit. Každý profil by měl mít své identifikující údaje (jméno, příjmení, věk) a měření, která byla pod tímto profilem provedena.

6.8.1 Filtrování měření

Měření, vyskytující se v jednotlivých profilech, by měla jít filtrovat podle daných parametrů. Mezi tyto parametry patří typ měření, rozsah délky měření, rozsah dat měření či konkrétní dny a nakonec hodina dne, ve které měření probíhalo.

6.9 Uspořádání dat a úložiště

Všechna data by měla být uspořádána podle profilové hierarchie a uložena na mobilním zařízení. Pro implementaci byly zváženy dva typy úložišť, přesně řečeno SQLite a FileSystem, tudíž lokální databáze a souborové úložiště v mobilu. Jednou z daných podmínek bylo však ukládání do CSV souboru, blíže popsaného v sekci 5.6, a přístupu k datům mimo aplikaci, tudíž jedinou volbou je použít FileSystem.

Samotná hierarchie FileSystemu aplikace se bude logicky podobat co nejbližší profilové hierarchii, tudíž každý profil má svou vlastní složku, jejíž obsahem budou všechna měření a soubor, shrnující všechna měření s dodatečnými informacemi jako například délka měření.

6.9.1 Import a export dat

Import a export se dá řešit s nebo bez pomoci internetu (například pomocí bluetooth). Obě tyto možnosti poskytuje komponenta Androidu *Intent*, která dokáže najít, otevřít a pracovat s aplikacemi, nainstalovanými v systému. Data lze poslat po kouscích či pomocí souborových formátů pro kompresi dat (ZIP, RAR). Nejlepším řešením pro rychlost a kompaktnost bude právě kompresní formát ZIP, který je také součástí platformy Java.

6.10 Vizualizace dat

Mezi povinné funkce aplikace patří zpětné zobrazení naměřených dat a analýz, provedených na těchto datech. Pro tuto funkčnost se nejvíce hodí graf. Jelikož Android nemá své vlastní zabudované knihovny pro tvoření grafů, tak vzniklo mnoho Open-source knihoven, které poskytují právě tvoření grafů. Samotné knihovny zde popisovat nebudu, avšak pro zájemce zde nechám referenci [32], která obsahuje seznam knihoven pro tvoření různých typů grafů. Jelikož by bylo vhodné při měření zobrazovat tento graf v reálném čase, tak jsem se rozhodl použít knihovnu *GraphView* [33].

6.11 Vizualizace měření

Pro měření bez podnětů je možno nechat obrazovku vypnutou, avšak bylo by vhodné, například u biofeedbacku, data zobrazovat tak, aby na ně mohl uživatel reagovat. Proto bude použita dříve zmíněná knihovna *GraphView*.

Pomocná vizualizaci při měření zvuku lze zařídit například pomocí vykreslení jeho frekvencí, avšak pro zajištění přesných reakcí na podněty zvuku je nejlepší, aby si uživatel vybavoval věci sám, bez jakékoliv vizualizace, proto vizualizace této části není nutná.

Vizualizace videí i obrázků lze zařídit pomocí pomocí zabudovaných Android komponent. Pro obrázkové měření je také nutno zajistit, aby se obrázky ukazovaly jako prezentace. Mezi jednotlivými snímky by také měla být možnost pauzy (černá obrazovka), aby se lépe zachytila reakce uživatele na jednotlivé obrázky.

6.12 Komunikace se Sensatio zařízením

Jelikož potřebujeme měřit po dobu určitou i neurčitou, tak je nutné se pokusit vyhnout nucenému ukončení (viz. 4.4.2). Potřebujeme také zaručit, aby se připojené zařízení neodpojilo při změně aktivity. Řešením jsou právě služby, které umožňují provádění dlouhodobých operací v pozadí. Pro naše potřeby a podle [21] je nejvíce efektivní použít službu v popředí.

6.13 Návrh UI

Pro vzhled UI jsem použil design *AppCompact*, který je zabudovaný do platformy Android.

6.13.1 Menu

Menu je základem UI aplikace a bude proplétat všechny části k sobě. Pro tuto funkčnost nám bude stačit Android Toolbar, který poskytuje klasický panel na vrcholu stránky s menu navigací. V případě potřeby je do něj také možno vložit tlačítka.

6.13.2 První spuštění aplikace

Jelikož je velká pravděpodobnost, že aplikaci a zařízení Sensatio budou používat lidé, kteří s tímto zařízením nemají zkušenosti, tak je nutné těmto uživatelům ukázat, jak se pracuje s aplikací. Řešením je příručka, která se ukáže při prvním, případně dalším, spuštění aplikace a provede uživatele tím, jak nastavit parametry a provést první měření.

6.13.3 Hlavní stránka

Hlavní stránka je navržena pro urychlení každodenního měření. Měla by poskytovat opětovné připojení na zařízení a různé možnosti měření. Pro zvolení parametrů měření se dá využít dialog.

6.13.4 Bluetooth

Bluetooth část by měla poskytnout rozhraní pro nalezení a připojení Sensatio zařízení.

6.13.5 Profily

Část s profily by měla poskytovat rozhraní pro vytvoření uživatele. Jednotlivé profily by pak měly mít své vlastní rozhraní, obsahující přehledy o profilu a jejich měření. Dále by měla být poskytnuta možnost jednotlivé profily exportovat či smazat.

6.13.6 Přehled měření

Přehled měření by se měl ukázat při vybrání měření v profilu uživatele a po jednotlivých měřeních. Přehled by měl obsahovat grafy s měřeními a analýzami společně s jejich nastavením.

6.13.7 Měření

Měření by mělo poskytnout možnost ukazování obrázků, videa, audia a grafu. Pro zajištění nekonečného měření by také měly existovat tlačítka, které zajistí spuštění, pozastavení či úplné zastavení měření.

6.13.8 Nastavení

Pro implementaci nastavení lze využít speciální rozhraní, poskytnuté Android studiem, pro nastavení různých částí aplikace.

Kapitola 7

Implementace

Tato kapitola se zabývá implementační částí aplikace a bude zde popisovat všechny implementované funkce aplikace.

7.1 Globalizace

Jak už bylo zmíněno v návrhu, bylo by vhodně poskytnout více jazyků pro české a jiné uživatele. Původně jsem chtěl, aby si jazyk uživatel měnil sám přímo v aplikaci, avšak kvůli abnormálnímu chování během změny orientace aplikace a změn aktivit jsem se rozhodl jí implementovat jiným způsobem.

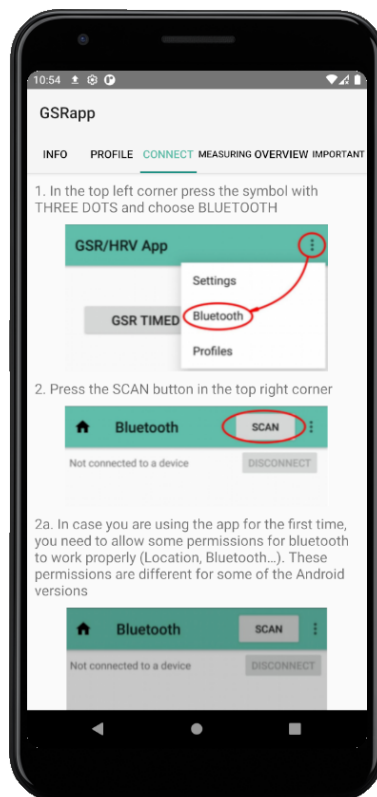
Vybraný jazyk aplikace tedy funguje podle toho, který jazyk má uživatel vybraný na svém mobilním zařízení. Řešení je poměrně jednoduché a využívá také oddělení souborů s kulturou od zbytku aplikace. V aplikaci jsou vytvořeny dva XML soubory, které obsahují všechny řetězce využívané v aplikaci. Jeden XML soubor (*strings.xml*), který lze v projektu nalézt pod *values*, obsahuje výchozí jazyk aplikace (angličtina), který je vybrán v případě, že uživatel na mobilu nemá nastavenou češtinu. V opačném případě se využije druhý soubor řetězců, nalezen pod *values-cs*, a jazyk je nastaven na češtinu.

Řetězce ve zmíněných XML souborech musí být stejně pojmenovány a v aplikaci se pak vypíší pomocí metody *getString*.

7.2 První spuštění aplikace

Aby jsme pomohli uživateli orientovat se v aplikaci, tak byla vytvořena příručka, která vysvětluje důležité části aplikace. Tato příručka je součástí aplikace a ukazuje se hned po prvním spuštění aplikace. Aktivita s částmi příručky je řešena pomocí fragmentů s možností pohybovat se mezi jednotlivými fragmenty prstem. Jednotlivé části vysvětlují jak si vytvořit nový profil, jak se při-

pojit na zařízení, jak nastavit a spustit první měření, jak exportovat data a nakonec jak si zpětně prohlédnout měření. Vzhled aktivity, implementující přehled příručky lze vidět na obrázku 7.1.

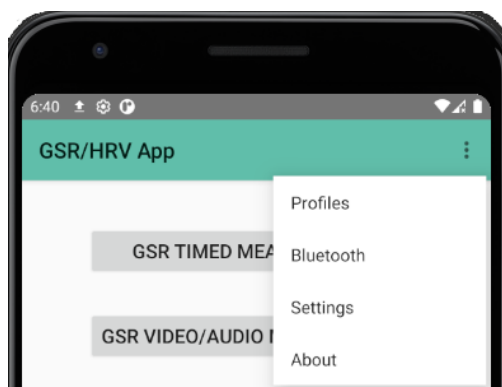


Obrázek 7.1: Vzhled příručky v aplikaci

7.3 Navigace

Navigace či menu, jak už bylo zmíněno, je důležitou součástí systému, která poskytuje rychlý přístup ke všem částí aplikace. Základní menu se skládá ze tří teček (Android kebab menu design), které se po kliknutí rozbálí a nabídnou možnost navigace na různé aktivity, v našem případě to jsou Bluetooth, Profily, Nastavení a O aplikaci. Menu je implementováno pomocí Android *Toolbar* a lze ho vidět na obrázku 7.2.

Jelikož se v aplikaci tento toolbar využívá vícekrát, tak je vhodné, pro lepší přehlednost kódu, obalit všechny třídy třídou, která poskytne zabudování menu a zajištění jeho chování ve všech aktivitách. Většina aktivit, které používají toto menu, tedy rozšiřuje třída *BaseActivity*, kde se nachází kód, který zajišťuje inicializaci a chování po kliknutí na jednotlivé položky menu. Aby se ale daly využít základní metody životního cyklu v aktivitách, tak je nutno tuto třídu rozšířit třídou *AppCompatActivity*.



Obrázek 7.2: Toolbar a rozbalené menu

7.4 ViewModely a změna orientace mobilu

Jak už bylo vysvětleno v návrhu, rozhodl jsem se pro implementaci možnosti změny orientace ve většině aktivit. Aby se uložily stavy či proměnné, držící data jednotlivých aktivit, tak se dá využít *ViewModel* (viz. 4.4.1). Viewmodely v naší aplikaci podporují většinu aktivit, jelikož její schopnost vědomí o životním cyklu je velmi užitečná a může udržet data přes změny orientace mobilního zařízení či přepnutí na jiné aplikace.

Aby se dal viewmodel použít, je nutné vytvořit třídu rozšiřující *ViewModel*. Tento viewmodel pak lze spojit s aktivitami pomocí *ViewModelProvider.get*, jejímž parametrem je právě ta třída, která implementuje viewmodel. Vracený objekt je vytvořená instance viewmodel třídy a může se díky němu vyvolávat metody daného viewmodelu.

7.5 Bluetooth

Hlavní součástí aplikace je připojení se k poskytnutému zařízení Sensatio pomocí aplikace a BLE. Aby jsme se vůbec dostali k datům tohoto zařízení, je nutné se řídit danými kroky, kterými jsou vyhledávání Bluetooth zařízení, připojení se na něj a získání dat. Všechny tyto kroky jsou řešeny pomocí rozhraní pro připojení zařízení v aplikaci a služby.

7.5.1 Implementace pomocí služby

Služba v popředí se v systémech s verzí OS menší než 8.0 dá spustit pomocí metody *startService* a pro zařízení s verzí vyšší či rovnou 8.0 metodou *startForegroundService*. Samotné spuštění ale nezaručuje, že je služba v popředí. Pro posunutí služby do popředí je nutno vyvolat *Service.startForeground* v metodě *onStartCommand*, která se provede po spuštění služby. Taktéž je nutné deklarovat v *AndroidManifest.xml* povolení *FOREGROUND_SERVICE*.

V novějších zařízeních je také nutno ukázat notifikaci. Objekt notifikace je předáván jako parametr metodě *startForeground*. Aby se vůbec tato notifikace ukázala, tak je nutno na Androidu verze 8.0 a vyšší vytvořit kanál notifikací pomocí metody *NotificationManager.createNotificationChannel*.

7.5.2 Komunikace mezi službami a aktivitami

Jak bylo dříve zmíněno v sekci 4.4.3, je pro komunikaci mezi službou a aktivitami možno využít vázání. Vázání funguje na bázi klienta a serveru. Pro navázání je nutno využít rozhraní *IBinder*. Pro komunikaci jsem využil třídu *Messenger*, která komunikuje na bázi zpětného volání, anglicky známo jako *Callback*. *Handler*, vytvořený na straně serveru, dokáže přijmout callbacky, které byly vytvořeny klientem. Aby klient věděl, komu má tyto callbacky posílat, tak se využívá zabudovaná metoda, která vrací objekt s implementovaným rozhraním *IBinder*. Pro posílání zpráv lze využít metody *Messenger.send*, která jako parametr přijímá zprávu s daty (*Bundle*).

Komunikace mezi klientem a servisem je při použití messengeru jednosměrná. V našem případě ale bylo potřeba zajistit dvousměrnou komunikaci. Ta se dala vyřešit obdobným způsobem jako předchozí navázání komunikace. Komunikace služby s aktivitou se tedy dá zajistit lehce díky již dříve vytvořenému rozhraní pro komunikaci aktivity se servisem.

7.5.3 Bluetooth oprávnění

Pokud chceme vyhledávat zařízení a používat BLE, je nutné udělit několik povolení. V Androidu existuje více druhů povolení každý s jinou úrovní ochrany. Všechna povolení se musí deklarovat v manifestu. Povolení označené jako nebezpečné musí uživatel povolit sám v případě, že má verzi Androidu vyšší než 6.0. V případě nižších verzí se povolení udělují během instalace aplikace.

Systém Android poskytuje předem připravené dialogy pro potvrzení těchto povolení. Dialogy lze vyvolat za pomoci metody *requestPermissions*, která přijímá řetězec s chtěnými povoleními.

Pro možnost vyhledávání zařízení je nutné udělit nebezpečné povolení lokace, závislé na aktuální verzi systému Android. Tyto povolení jsou *COARSE*, *FINE*, *BACKGROUND*. Úroveň normální mají povolení *BLUETOOTH*, *BLUETOOTH_ADMIN*.

7.5.4 Vyhledávání zařízení

Pro vyhledávání je nutno použít třídu *BluetoothAdapter*, která se dá získat pomocí *BluetoothAdapter.getDefaultAdapter*. Tato třída poskytuje několik metod pro vyhledávání zařízení. V naší aplikaci se zařízení vyhledávají přes metodu *BluetoothAdapter.startDiscovery*.

7.5.5 Připojení

Pro připojení je potřeba, aby byl zapnutý Bluetooth a uživatel udělil oprávnění. Proces připojení lze vidět ve schématu C.1. Nutnou podmínkou provedení připojení v daném schématu je, že

se uživatel musí vyskytovat v Bluetooth aktivitě, která se již provázala se servisem (viz. 7.5.2). Dále pak je nutno znát fyzickou adresu vyhledávaného zařízení. Ta se dá získat pomocí metody *BluetoothAdapter.getRemoteDevice*. Mobilní zařízení se pak pokusí připojit přes metodu *BluetoothDevice.connectGatt*, jejíž hlavní parametr, který také zajistí, zda se zařízení připojilo či nikoliv, je *BluetoothGattCallback*. Pokud se zařízení připojilo, tak je nutné najít potřebné služby a charakteristiky. Důvod tohoto hledání je popsán v kapitole použitých technologií (5.7). Služby se ze zařízení dají získat pomocí metody *BluetoothGatt.discoverServices*, která vyvolává vytvořený callback. Výsledkem jsou pak všechny služby, které byly nalezeny na připojeném zařízení. Objekt nalezené služby je uložen a připraven k použití, tudíž k měření a získávání dat.

7.5.6 Získání UUID

K připojení na správné zařízení je nutno znát UUID charakteristiky, které by měly být součástí dokumentace Sensatio zařízení. Tuto dokumentaci jsem bohužel nenašel ani nedostal, tudíž jsem si musel najít UUID těchto charakteristik sám. Jako nástroj k rozpoznání charakteristik jsem použil aplikaci *Bluetooth LE Scanner* [34], která má rozhraní ke zjištění a exportu služeb a charakteristik připojených zařízení. Tato aplikace bohužel nerozpoznala názvy služeb ani charakteristik, tak jsem měl k dispozici pouze jejich UUID. Na Sensatio zařízení se nachází pouze jediná charakteristika, která má dvě hodnoty (Tonic a Phasic viz. 3.2), které se jako jediné neustále měnily. UUID této charakteristiky jsem pak využil v aplikaci během implementace Bluetooth služby.

7.5.7 Získávání a vysílání dat

Data se dají získat z nalezené charakteristiky pomocí metody *BluetoothGatt.readCharacteristic*, jejíž výsledek opět přijímá *BluetoothGattCallback*. Pomocí metody *BluetoothGattCharacteristic.getValue* lze získat data z charakteristiky. Tyto data jsou ve formě bajtů a jsou uložena v poli. V našem případě jsou v tomto poli dvě byte hodnoty (Tonic a phasic), jak už bylo dříve zmíněno v sekci 7.5.6. Data jsou původně celočíselné hodnoty (int), tudíž je nutné je převést.

Převedené hodnoty se pak vysílají do aplikace pomocí metody *sendBroadcast*, která přijímá intent se získanými daty jako parametr. Data vyslány tímto způsobem je možno v aplikaci zachytit kdekoli pomocí *BroadcastReceiver*. Metoda callbacku zavolaná po přečtení charakteristik, převod hodnot a následné vysílání lze vidět v kódu 7.1.

```
final BluetoothGattCallback bluetoothGattCallback =
    new BluetoothGattCallback() {
        @Override
        public void onCharacteristicRead(final BluetoothGatt gatt, final
            BluetoothGattCharacteristic characteristic, int status) {
            super.onCharacteristicRead(gatt, characteristic, status);
            byte[] value = characteristic.getValue();
```



```

        int value1 = (value[0] & 0xFF);
        int value2 = (value[1] & 0xFF);
        Intent valuesIntent = new Intent(ACTION_SEND_VALUES);
        valuesIntent.setFlags(Intent.FLAG_RECEIVER_FOREGROUND);
        valuesIntent.putExtra(ExtraConstants.GSR_VALUE1, value1);
        valuesIntent.putExtra(ExtraConstants.GSR_VALUE2, value2);
        valuesIntent.putExtra(ExtraConstants.GSR_TIME_VALUE, tickCounter);
        sendBroadcast(valuesIntent);
    }
};

```

Listing 7.1: Kód převodu a vysílání dat

7.5.8 Bluetooth UI

Skenování a připojení a odpojení zařízení je možno provést v Bluetooth aktivitě. Vzhled této aktivity lze vidět na obrázku 7.3. Povolení jsou vyvolána po zmáčknutí tlačítka skenovat. Když uživatel udělí povolení a spustí Bluetooth, tak může spustit skenování, které naplní seznam nalezenými zařízeními. Kliknutím na chtěné zařízení v seznamu, v našem případě *WBBracelet2*, umožní jeho připojení.

7.6 Umístění dat ve fyzickém úložišti mobilního zařízení

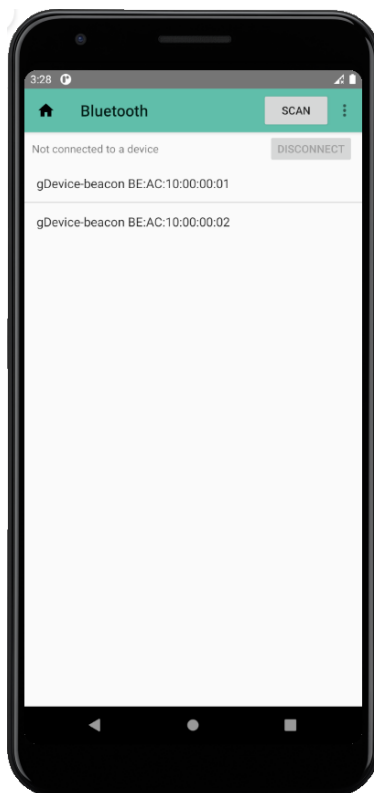
Jak bylo zmíněno, byla navržnuta hierarchie profilů, kterou je nutné replikovat do FileSystemu. Všechny data jsou uložena do externího úložiště mobilu, které se dá získat pomocí metody *Context.getExternalFilesDir*. Jako externí úložiště se v Androidu počítá buď externí paměťová jednotka (SD karta) a nebo Androidem zabudovaný FileSystem, který se využije pokud uživatel nemá SD kartu. Toto externí úložiště je volně přístupné a lze se k němu dostat přes jakýkoliv správce souborů. Externí složka je označena podle ID aplikace, tudíž v našem případě *com.example.gsrapp*.

7.7 Zpětná vazba

Zpětná vazba je v aplikaci zařízena popisem jednotlivých nastavení a dialogů, za pomoci titulků a textových boxů či ukazováním notifikace například při připojení k zařízení (viz. 7.5.1).

Dále pak ukázáním malého vyskakující okna (*Toast*) většinou při reakci aplikace na nějakou událost jako například připojení zařízení. Toto okno se dá vytvořit pomocí příkazu *Toast.makeText* a poté jeho zobrazení pomocí *Toast.show*.

Dále jsou součástí aplikace zvukové notifikace, které se přehrávají například při náhodném odpojení zařízení. Jednotlivé zvuky lze vytvořit za pomoci třídy *SoundPool* a její metody *SoundPool.load*, která načítá zvuky ze složky *raw*.



Obrázek 7.3: Vzhled Bluetooth aktivity

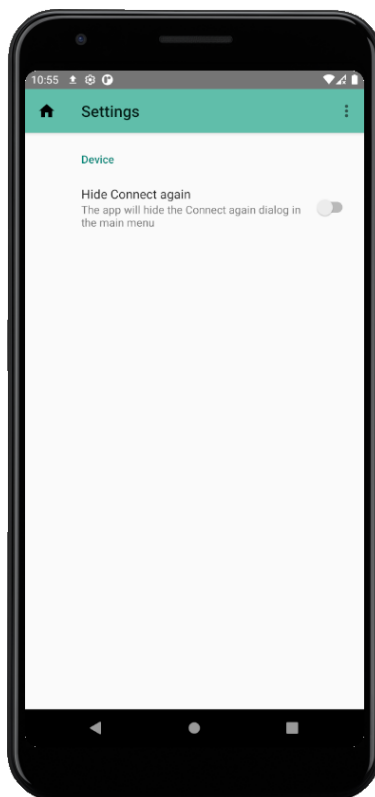
7.8 Nastavení

Jak už bylo zmíněno v návrhu, Android poskytuje přednastavenou komponentu nastavení pojmenovanou *Preference*. Tato komponenta poskytuje menší komponenty (Views), které se dají použít pouze v této aktivitě jako například různé přepínače či tlačítka. Tato přednastavená komponenta zajišťuje ukládání jednotlivých nastavení do *SharedPreferences*, které se pak dají získat pomocí metod pro práci s tímto úložištěm jako například *SharedPreferences.getBoolean*.

V naší aplikaci těchto nastavení nebylo moc potřeba, avšak pro nastavování věcí od synchronizace dat s databází po změnu vzhledu aplikace je toto řešení nejefektivnější. Vzhled této aktivity lze vidět na obrázku 7.4.

7.9 Hlavní stránka

Na hlavní stránce se dá dostat na jednotlivé měření pomocí tlačítek. Tlačítka mají zabudovanou kontrolu, zda je zařízení připojeno. Pokud není, tak se ukáže *Toast*, který nabádá uživatele, aby se připojil k měřicímu zařízení. Vzhled této aktivity lze vidět na obrázku 7.5.



Obrázek 7.4: Vzhled stránky s nastavením

Hlavní stránka také obsahuje možnost rychlého připojení měřicího zařízení, které bylo naposledy použito mobilem uživatele. Ukládá se fyzická adresa (MAC adresa) zařízení do *SharedPreferences*. Uložení této adresy inicializuje připojení měřicího zařízení, které má potřebné charakteristiky pro měření GSR. Pro zajištění připojení se opět používá komunikace pomocí vazání služby a aktivity.

7.10 Vizualizace pomocí grafů *GraphView*

Graf z knihovny *GraphView* [33] podporuje čárový, sloupcový a bodový graf. Jeho další schopností je jednoduché vykreslování dat v reálném čase. Dají se nastavit popisy jeho os, minimální a maximální X a Y osy a jiné. Pro vykreslování se používají jeho série, tedy propojené části grafu a datové body. Pro realizaci aktualizace dat grafu se používá *.appendData* metoda, která se volá na vytvořené série. Výsledek takového tvoření grafu lze vidět na obrázku 7.8.

7.11 Měření

Jelikož se data ze servisu posílají přes *sendBroadcast*, jak již bylo zmíněno v sekci 7.5.7, tak je pouze nutné zaregistrovat *BroadcastReceiver*. Časté přijímání dat v metodě *onReceive* pomůže posunout



Obrázek 7.5: Vzhled hlavní stránky s možností rychlého připojení

aktivitu měření na nejvyšší prioritu (viz. 4.4.2) a tím zajistit, že systém aplikaci nezničí, jelikož dělá důležité operace. Aby jsme zajistili, že se takhle bude dít i při přepnutí aplikace, tak jsem broadcastreceiver registroval v metodě *onCreate*, tedy při stavbě aktivity a odregistroval v metodě *onDestroy*.

Tato aktivita je pro zredukování opakování kódu implementována v jedné třídě. Aktivita opět komunikuje s viewmodelem, který pak dále komunikuje s částmi aplikace, určené pro ukládání dat (repositories).

Měřicí aktivita také jako jediná nemá možnost změny orientace, jelikož se během této změny ničí aktivita 4.4.1, čímž se se odregistrouje broadcastreceiver a data jsou ztracena.

7.11.1 Nastavení měření

Součástí každého měření je jeho nastavení. Nastavení lze spustit, pokud uživatel připojil své mobilní zařízení k měřicímu zařízení Sensatio. Nastavení má společnou část, která se skládá z vybrání profilu, pro které bude měření provedeno, a vybrání rychlosti čtení dat.

V časovém, zvukovém, obrázkovém a video měření se také nastavuje doba normalizace Sensatio dat. Tato doba je implementována stylem časovače, implementovaného pomocí třídy *CountDownTimer*. Jeho chování jsem implementoval ve své vlastní třídě pomocí návrhového vzoru *Singleton*.

V měření s videi, zvuky a obrázky je nutno vybrat ze systému odpovídající zdroje, které se během měření budou přehrávat a ukazovat. Pro obrázky se zde nachází i nastavení prezentace, tudíž rychlost přechodu snímků a zda-li se má použít přechodný obrázek (viz. 6.11).

V nekonečném měření lze určit poznámky k měření a jeho název.

7.11.2 Počátek měření

Po nastavení lze spustit aktivitu měření, která z počátku spouští časovač, z důvodu normalizace hodnot v měřicím zařízení. Během tohoto času se uživateli nabízí dialog pro vyplnění jeho pocitů před začátkem měření. Mezi tyto pocity patří aktivita (pasivní a aktivní), napětí (napjatý a uvolněný) a pohodlí uživatele (příjemný a nepříjemný). Pokud uživatel nestihne potvrdit dialog s pocity, tak se časovač spouští znova. Po úspěšném poznamenání svých pocitů do dialogu a po ukončení časovače může začít měření.

7.11.3 Průběh měření

Aktivita po začátku měření spojí s fragmentem, který obsahuje různé funkcionality podle typu měření.

Data se během měření posílají z implementované služby pomocí *sendBroadcast* a přijímají pomocí *BroadcastReceiver*. Data jsou ihned aktualizována a uložena pomocí třídy *MutableLiveData*, která je generická a přijímá implementovanou třídu pro všechna měření *MeasurementGSR*. Data této třídy mohou být sledována (návrhový vzor *Observer*) různými aktivitami pomocí metody *observe*. Aktualizaci dat v *MutableLiveData* tedy obdrží všechny aktivity, které sledují tuto třídu s daty.

Pro zaručení opakovaného zasílání dat je ve službě implementován *Timer*.

7.11.4 Vizualizace dat

Pro vizualizaci časového a nekonečného měření je použit graf knihovny *GraphView* (viz. 7.10) s možností aktualizace za reálného času.

Pro video i zvuk se používá komponenta *VideoView*, která automaticky nastaví orientaci mobilu podle orientace videa. Video nelze zastavit.

Pro obrázkové měření je nutno zjistit, jakou orientaci bude aktivita mít. To se dá určit při počítání obrázků, které mají orientaci na výšku či na šířku. Šířka i výška se dají zjistit pomocí třídy *BitmapFactory*, která přijímá URI načtených obrázků a dekoduje je pomocí *BitmapFactory.decodeStream*.

Pro zobrazování obrázků se používá komponenta *ViewPager*, která normálně umožňuje přechod mezi obrázky pomocí prstu, v naší aplikaci je ale tato funkce zakázána, protože chceme aby uživatel byl v klidovém stavu během měření. Obrázky se střídají s černou obrazovkou (klidovou obrazovkou), pokud je tak dáno v nastavení měření.

7.11.5 Doba po měření

Měření končí po ukončení videí, prezentace, zvuku či po vyprchání času. Tyto jednotlivé události pak spustí ukládání měření. Měření jsou ještě v paměti mobilu a tak je lze využít pro zapsání statistického souboru a vypočtení analýz (viz. 7.12).

Po měření se ukazuje graf a v případě časovaných měření i přehled měření a analýz (vice v sekci 7.14).

7.11.5.1 Ukládání měření

Jednotlivá měření jsou zapisována do *ArrayList<Integer>*. První dva listy obsahují phasic a tonic komponenty vodivosti kůže (viz. 3.2) a třetí obsahuje vypočtené hodnoty, které se používají v grafech. Výpočet těchto hodnot lze vidět ve vzorečku 7.1. Tyto listy jsou ukládány do externího uložení pomocí tříd *BufferedReader* a *FileReader*. Zapsané soubory mají svůj daný jmenovací systém (typ_datum_hodina.csv). Aby se správně napsal .csv soubor, tak je nutné všechny pole oddělit čárkou.

$$value = phasic + 256 * tonic \quad (7.1)$$

Měření se během nekonečného měření ukládají průběžně. V aplikaci je tedy implementováno automatické ukládání. Doba jednotlivých uložení je 5 minut.

7.11.5.2 Ukládání statistik

Aby se při zobrazení měření nemusely opakovaně vypočítávat statistiky, tak existuje soubor *statistics.csv*, který obsahuje všechna měření, které daný profil podnikl. Jeho součástí jsou základní informace o profilu a měření, tudíž ID (názvy souborů, později použité pro referenci na jednotlivá měření), rychlost čtení, čas vychlazení (normalizace), doba měření, pocity před měřením a analýzy.

7.12 Provedení analýzy na datech

V projektu se používá knihovna, poskytnuta Ostravskou univerzitou. Tato knihovna dokáže vypočítat amplitudu a silné a slabé vrcholy (peaks).

Má vlastní třída pro vypočtení analýz dokáže vypočítat tyto základní analýzy:

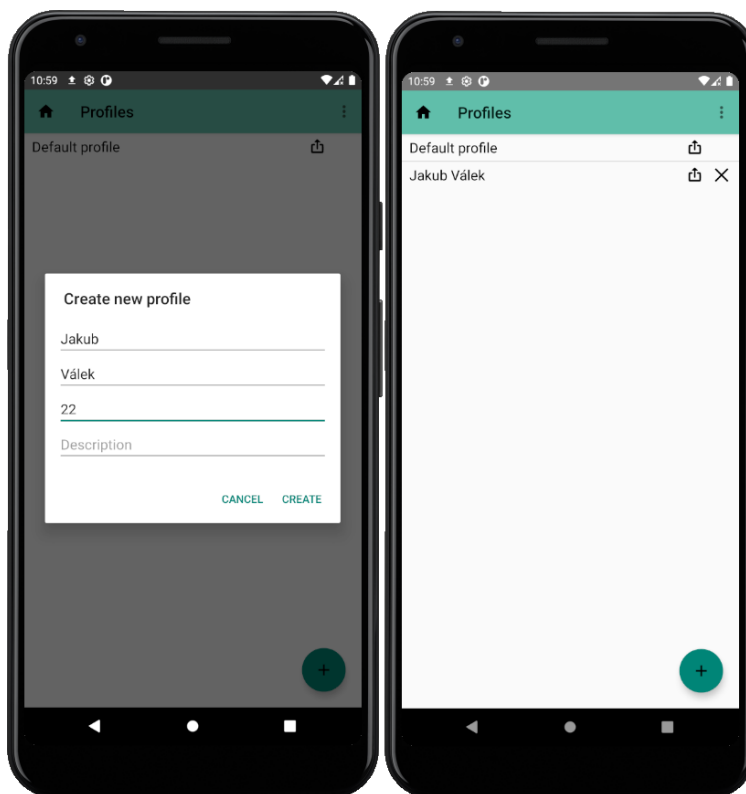
- Minimum
- Maximum
- Průměr
- Medián - střední hodnota

- Standardní odchylka (STD) - anglicky Standard Deviation, určující rozptyl datových hodnot vzhledem k jejím průměrům.
- Trend hodnot - změna v hodnotách na začátku a konci měření.

V obou případech je pro analýzu nutné poskytnout vektor zpracovaných dat (viz. 7.1), na kterých se provedou analýzy. Pro oba případy stačí zavolat pouze jednu metodu či konstruktor, pro získání všech analýz.

7.13 Profily

Pro rozdělení uživatelů, měřících na jednom zařízení, je vhodné implementovat profily. V aplikaci je implementován seznam, zobrazující všechny vytvořené profily. V případě, že se měří pouze jeden uživatel, je v aplikaci výchozí (default) profil, který se nedá smazat, aby vždy existovalo místo pro ukládání měření. V případě zájmu si může uživatel vytvořit svůj vlastní profil po stisknutí tlačítka plus. Vzhled tvoření profilu a samotné profilové aktivity lze vidět na obrázku 7.6. Tvoření profilů má povinné pole jména, příjmení a věku a nepovinné pole popisu. Profily je v aplikaci možno smazat a exportovat pomocí odpovídajících tlačítek (ikona křížku a exportu).

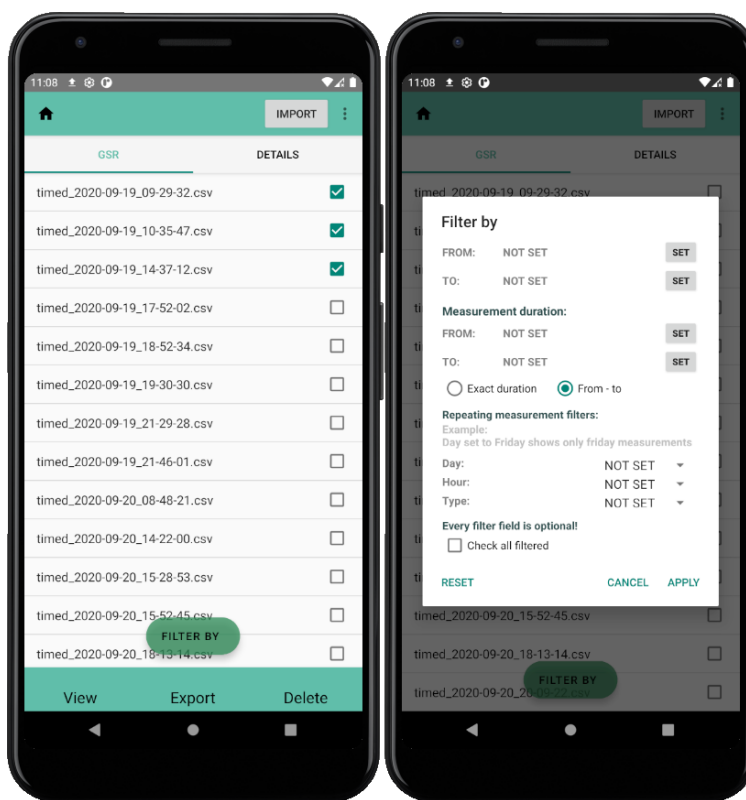


Obrázek 7.6: Vzhled tvoření profilu a stránky s profily

7.13.1 Detaily profilu

Po kliknutí na jakýkoliv profil lze zobrazit informace o něm. Mezi tyto informace patří detaily profilu (jméno, příjmení, věk, popis) a jeho měření, implementována pomocí seznamu s možností výběru více měření.

Samotná aktivita je implementována pomocí fragmentů s možností se pohybovat mezi jednotlivými detaily či měřeními pohybem prstu. Její vzhled lze vidět na obrázku 7.7.



Obrázek 7.7: Vzhled aktivity přehled profilu a dialogu pro filtrování měření

7.13.2 Ukládání profilů

Profily se v aplikaci ukládají do textového souboru jako objekty vytvořeného modelu pro všechny profily *LocalProfile*, jehož součástí je vlastník *Person* a pro zakázání mazání také vlajka, zda je profil výchozí či nikoliv. Pro jednotlivé profily je také vytvořena složka, do které se vkládají statistiky a měření provedené na těchto profilech. Složky mají tvar příjmení#ID, kde ID je číslo, které se inkrementuje při vytváření dalších profilů. Toto umožňuje aplikaci rozeznat uživatele v případě, že mají stejná příjmení.

7.13.3 Filtrování měření

Díky jmenovacího systému všech měření, zmíněného v sekci 7.11.5.1, není nutno načítat a číst žádné soubory, pouze jejich názvy. Názvy souborů se dají načíst pomocí třídy *File* a jejích metody *File.listFiles* a *File.getName*. Tyto názvy souborů jsou pak rozebrány (parsovány) na jejich jednotlivé části. Pomocí těchto částí se pak dají zajistit různé podmínky, které vyfiltrují měření.

Samotný filtr je implementovaný pomocí fragmentovaného dialogu, tudíž je schopný přežít orientaci mobilu oproti jiným dialogům. Dialog komunikuje se třídou pomocí vytvořeného rozhraní *MeasurementFilterDialogListener*. Filtr také nabízí možnost výběru všech měření a resetování všech nastavených filtrů.

Spuštění fragmentu pro filtrování zajišťuje v aplikaci tlačítko na spodní části stránky.

7.13.4 Exportování

V naší aplikaci lze exportovat celý profil, což lze z aktivity profilů a exportovat vybraná měření, což lze provést z přehledu měření na jednotlivých profilech. Exportovat lze dvěma způsoby, kterými jsou ZIP a pomocí Google drive rozhraní pro programování aplikací (API), anglicky taky známo jako Application Programming Interface.

7.13.4.1 Exportování pomocí ZIP

Java nabízí knihovnu pro práci s ZIP soubory pojmenovanou *java.util.zip*. Tato knihovna používá třídu *ZipEntry*, které zapisuje jednotlivé soubory do ZIP. Při konstrukci jejího objektu stačí cesta a název souboru získaného z příslušného *File* objektu. Jednotlivé zipentry objekty pak zpracovává *ZipOutputStream*, který píše jednu za druhou do ZIP souboru pomocí *FileInputStream*.

Výsledný ZIP soubor pak lze poslat pomocí Intentů do jakékoliv aplikace, implementující možnost přijmutí *Intent.ACTION_SEND* intentů. Příkladem takových aplikací může být FileSystem, discord, google drive a jiné.

Pro poslání intentu je nutné znát jednotný identifikátor zdroje (URI), z anglického Uniform Resource Identifier, který identifikuje dokument ZIP. Dále by měl intent vědět o tom, jakého typu je ZIP soubor. To se dá zajistit pomocí *Intent.setType*, kde typ bude v našem případě *application/zip*.

7.13.4.2 Exportování pomocí Google drive API

Google poskytuje API [35] pro práci s google drivem. Toto je rozdílné od exportování ZIP pomocí aplikace Google drive a to tak, že pro použití exportu na google drive není nutno použít žádné jiné nainstalované aplikace. Všechna měření exportována tímto způsobem jsou uložena do jediné složky na disku, kterou pak lze sdílet. Operace s touto API jsou asynchronní a pro správnou funkčnost je nutno registrovat aplikaci na stránce *console.cloud.google.com* pomocí SHA-1 certifikátů [35].

Proces exportu na google drive začíná přihlášením uživatele přes *GoogleSignInOptions*, kde se po pokusu přihlášení, jehož vzhled poskytuje samotná API, získá objekt *Task*. Na tento task se dají připsat posluchače (listeners) událostí, jako například *Task.addOnSuccessListener*, který určuje co se stane, pokud se task provede úspěšně. Pokud se uživatel úspěšně přihlásil, je možno začít stavět *Drive* a *GoogleDriveRepository* objekty, které se postarají o posílání přiložených souborů na google drive.

7.13.5 Importování

Import je možno zahájit pomocí tlačítka import v aktivitě přehled profilu. Jeho součástí je základní zpětná vazba informující uživatele o správném stylu importu.

Importuje se ZIP soubor, který má stejné vlastnosti jako exportovaný soubor. Tudíž je nutné, aby importovaný soubor byl typu ZIP a měl v sobě měření, dané jmenovacím systémem aplikace (viz. 7.11.5.1).

Pokud se při práci s daty někdy ztratil soubor *statistics.csv*, ve kterém jsou uloženy základní informace a analýzy každého měření, tak se import pokusí tento soubor sestavit sám a to načtením potřebných měření, pro které jsou znovu vypočteny analýzy, zjištěna rychlost čtení a délka měření.

7.14 Přehled měření a analýz

Analýzy se v této aktivitě získávají přímo ze souboru *statistics.csv*, kde byly vypočteny analýzy po měření nebo po importu měření.

7.14.1 Data pro přehled

Data pro graf, který ukazuje všechny měření v průběhu času, se získávají z jednotlivých souborů měření. Data pro grafy analýz lze získat z dříve zmíněného souboru se statistikami.

Grafy pro analýzy jsou časové řady, tudíž v každém grafu lze určit časový interval (například 2 hodinový interval), který pak shrne všechna měření v 24 hodinovém okně do těchto časových intervalů. V každém intervalu lze zobrazit minimum, maximum a průměr všech načtených analýz. Příkladem může být minimum z měření, pokud se načtou minima všech měření, tak lze určit minimum, průměr a maximum těchto minim. Jednotlivé části se pak ukážou v grafech a jsou mezi sebou spojeny. Výsledkem jsou 3 spojené série v každém grafu.

Pro jejich výpočet jsem implementoval vlastní třídu *TimeSeriesHelper*, která implementuje metodu *getTimeRangeViableStatistics*. Tato metoda bere v parametrech typ měření, statistiky a časový interval. Dále pak rozděluje všechny ID měření na části, tudíž komponenta času, data a typu měření. Hlavní komponentou pro rozdělení měření do intervalů je komponenta hodin společně se získanou časovou částí ID. Typ měření určuje typ analýzy (například minimum). Z metody se vrací list s třemi dříve řečenými vlastnostmi.

7.14.2 Typy přehledů

V přehledu po měření (aktivita) jsou načtena všechna měření, které splňují dané požadavky, tudíž všechna měření, která mají stejnou dobu normalizace, délku měření a rychlost čtení jako proběhlé měření. Společně s nově vypočtenými analýzami je cílem uživateli ukázat, kde se po daném měření nachází oproti jeho ostatních měření a to tak, že se v grafech objeví černý bod, který označuje proběhlé měření.

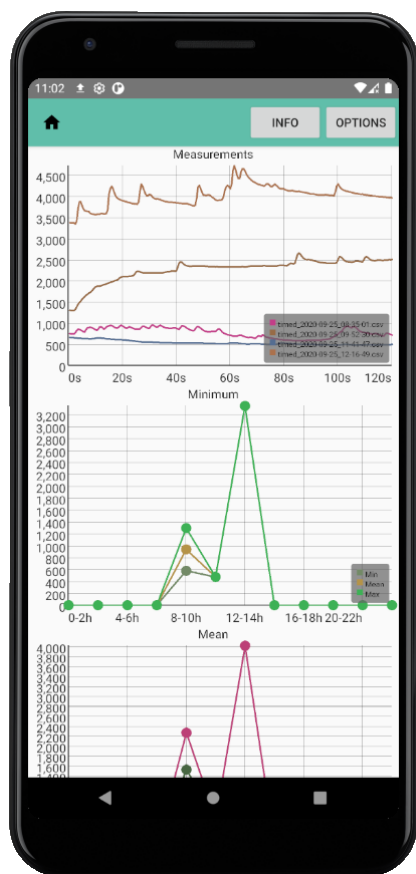
Při výběru měření v profilech se načítají všechna vybraná měření bez jakýchkoliv omezení. Analýzy jsou provedeny na všech datech stejně a je pouze na uživateli, co si chce zkusit zobrazit a analyzovat.

7.14.3 UI přehledu

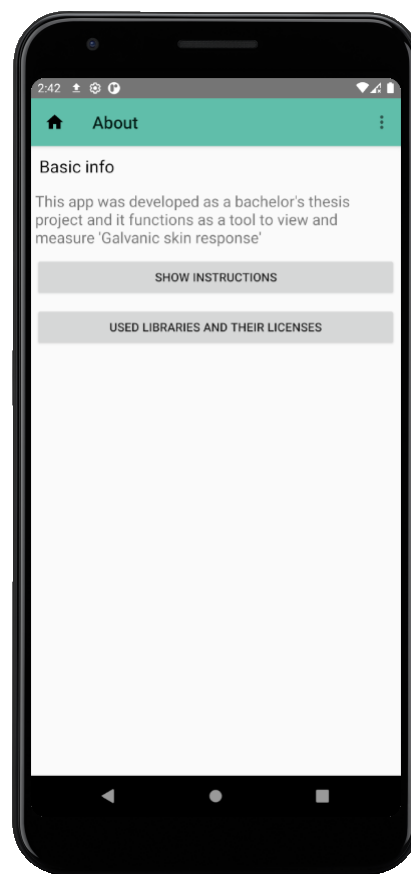
Pro přehled analýzy se opět implementuje knihovna *GraphView*. Aktivita s těmito měřeními je složena s několika fragmentů, závislých na počtu analýz k zobrazení. Aktivita implementuje *OptionsListener*, který odposlouchává jednotlivé fragmenty, díky čemuž se dá měnit barva grafů, interval hodin a určovat, zda zobrazovat legendu či nikoliv. Dialog pro změnu těchto vlastností spouští tlačítko v toolbaru (options/možnosti). V přehledu lze taky zobrazit jednotlivé statistiky opět pomocí tlačítka v toolbaru (info), které načte tabulku s danými statistikami. Vzhled této aktivity lze vidět na obrázku 7.8.

7.15 O aplikaci

Poslední částí aplikace je pomocná aktivita pojmenovaná O aplikaci. Její vzhled lze vidět na obrázku 7.9. Tato aktivita umožňuje zobrazení příručky a jednotlivých knihoven a jejich licencí. Licencí jsou vygenerovány knihovnou *com.google.android.gms:oss-licenses-plugin*, která také poskytuje UI pro prohlížení daných licencí.



Obrázek 7.8: Přehledy měření a analýz



Obrázek 7.9: O aplikaci

Kapitola 8

Testování aplikace

V této kapitole bych chtěl popsat, jak se testovala aplikace během vývoje. Dále bych chtěl popsat experimentální část, která povídá o práci s grafy, informace o měřeních a jiné. Pro zajištění testování lze využít tvorbu *signed .apk* souborů, sloužící k instalaci aplikace. Tuto funkcionalitu poskytuje vývojové prostředí Android studio.

8.1 Testování aplikace během vývoje

Aplikace měla v průběhu vývoje několik verzí. Každá verze pomalu přidávala jednotlivé funkce. Testování bylo z velké části na mé straně. Android studio poskytuje AVD 5.4, kterým se dá ladit přímo na počítači. AVD, na kterém jsem testoval některé funkce, běželo na verzi 10.0 (Q). Bohužel AVD chybí funkce pro připojení na zařízení Sensatio, tudíž bylo nutné testovat tuto funkčnost na mobilních zařízeních. Testování tedy probíhalo na mém zařízení, běžící na verzi 5.0 (Lollipop), které mi umožnilo testování na starších zařízeních.

Druhá část testování byla na straně mého vedoucího, který má mobil verzi 10.0 (Q), a na straně koncových uživatelů, kteří si mohli aplikaci a její nové verze volně stáhnout z Google drive. Díky jejich testování jsem získal zpětnou vazbu o funkčnosti aplikace na novějších zařízeních.

8.2 Experimentální část

Aby jsem získal data pro analýzy, tak jsme se já i můj vedoucí bakalářské práce v průběhu vývoje měřili pomocí zařízení Sensatio. Získaná data na obrázcích jsou naměřeny v září od pana vedoucího (obrázek 8.2) a moje z října (obrázek 8.3). Všechny data byla naměřena se stejnými parametry, tudíž 500 milisekundové čtení, 60 sekund normalizace zařízení a měření probíhala 2 minuty.

8.2.1 Význam dat

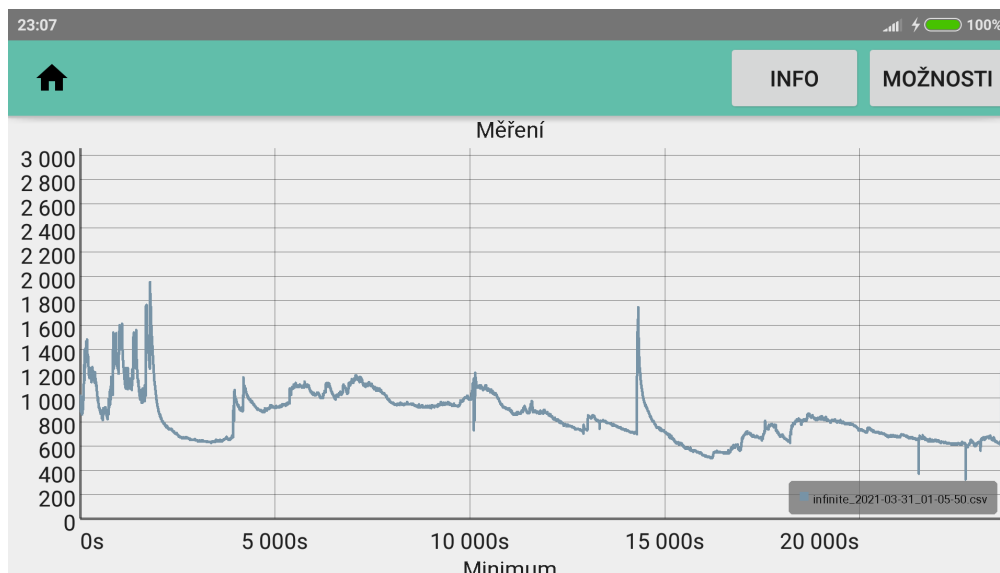
Pokud se podíváme na průměrnou hodnotu všech dat tak lze vidět, že pan vedoucí má vyšší hodnoty. To může znamenat mnoho věcí, byl ten měsíc více vystresovaný nebo zaneprázdněný a tím se zvýšilo jeho pocení, což způsobilo naměření větších hodnot. Co se týče mých průměrných hodnot, tak ty jsou výrazně nižší. V říjnu již probíhal začátek zimního semestru, tudíž nebylo moc stresu. Pokud by měření proběhlo například ve zkouškovém období, tak by mé hodnoty dost pravděpodobně překročily ty od pana vedoucího.

Dále je zajímavé se například podívat na trend hodnot, který určuje jak moc velký rozdíl byl mezi první a poslední naměřenou hodnotou. Mé měření průměrně klesala a vím, že jsem si většinou uvědomil, že právě měřím a díky toho jsem se dokázal uklidnit a zmírnit tak své pocení. Na rozdíl od pana vedoucího, jehož hodnoty průměrně stoupaly při většině měření.

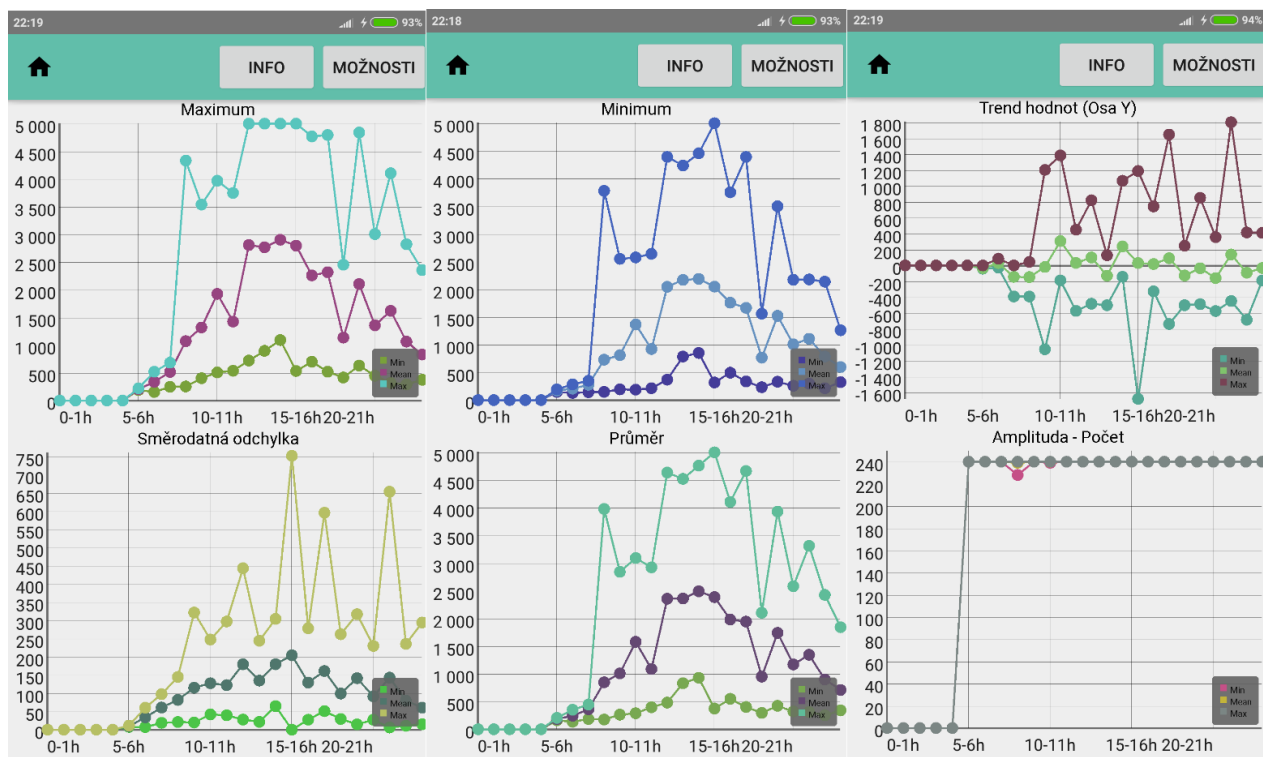
Všechny tyto data a analýzy se dají ve výsledku použít jako biofeedback a lze se pak například uklidnit během měření, pokusit se nestresovat během dne či zjistit stresové hodiny během dne a jiné.

8.2.2 Nekonečné měření

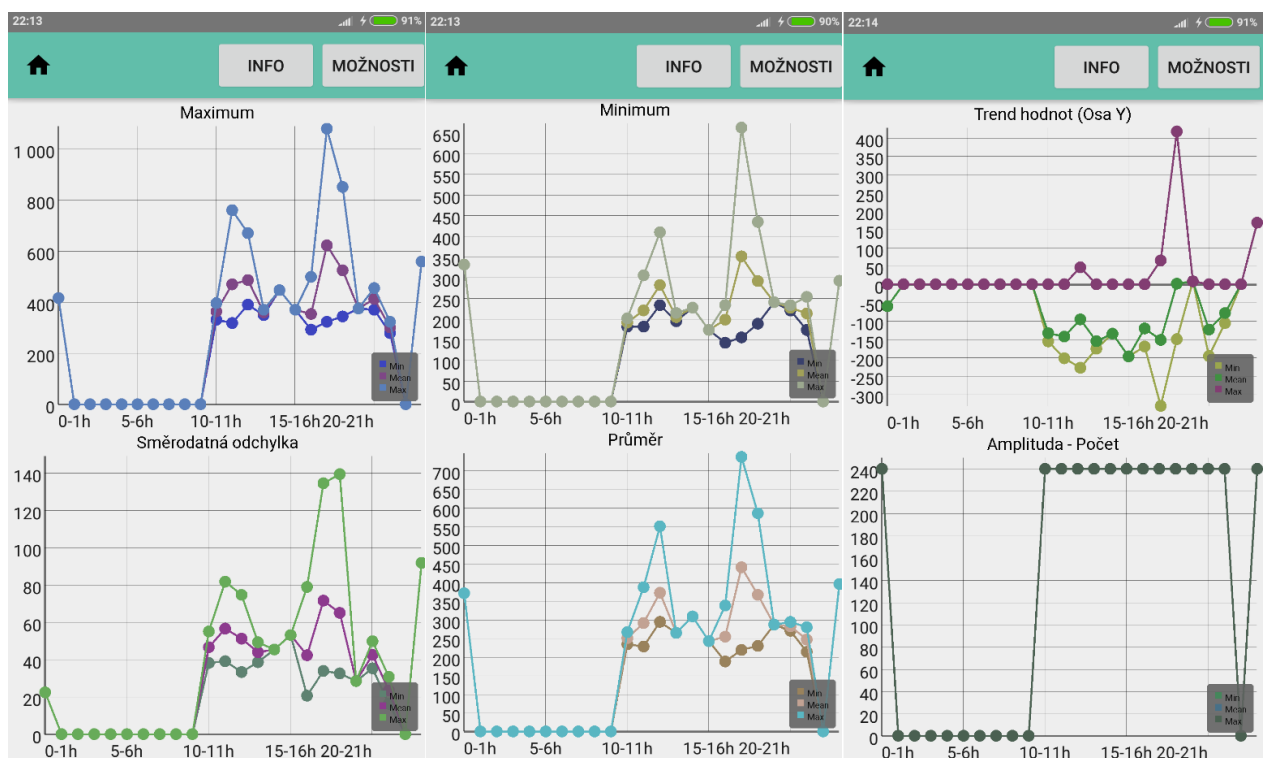
Jako příklad nekonečného měření jsem se rozhodl změřit se během noci. Od zařízení jsem byl během noci většinou jeden metr, šel jsem spát kolem půlnoci a vstával jsem přibližně o osm hodin později. Výsledek tohoto měření lze vidět na obrázku 8.1. Na datech lze většinou vidět, kdy jsem se otáčel či trochu svlékl náramek. Pro správnou analýzu jednotlivých vrcholů by bylo nutno natočit video mého spánkového cyklu během měření.



Obrázek 8.1: Měření přes noc



Obrázek 8.2: Přehled měření pana vedoucího ze září



Obrázek 8.3: Přehled mých vlastní měření z října

Kapitola 9

Závěr

Cílem této bakalářské práce bylo vytvoření Android aplikace, seznámení se s platformou Android a pochopení principu poskytnutého zařízení Sensatio a jeho připojení na mobilní zařízení.

Při implementaci aplikace bylo důležité si uvědomit, jak se chová zařízení Android například při změny orientace či přepnutí na jinou aplikaci. Řešením byly právě viewmodely, které pomohly zachovat stavy během těchto změn a rozdělit aplikaci do menších částí, tudíž rozdělit tím logiku prezentační vrstvy od vrstvy datové a aplikační.

Výsledkem byla aplikace ulehčující měření elektrodermální aktivity. Mezi implementované funkce aplikace se řadí připojení se na měřicí zařízení, získání dat z měřicího zařízení, různé druhy měření, přehledy měření a vypočtených základních analýz na časových řadách.

Aplikace byla ze strany koncových uživatelů úspěchem a bude se dále vylepšovat. Cílem aplikace v budoucnu bude poskytnutí platformy pro zredukování denního stresu pomocí meditace, tedy pomocí reakcí na zvuky, videa a různých jiných uklidňujících prvků. Tudíž do měření se budou například přidávat zvuky, které budou upozorňovat na zvyšující se GSR hodnoty. Aplikace se tedy dá dále rozšiřovat pomocí různých biofeedback technik.

Mezi vylepšení architektury či práci s daty můžeme zařadit například nahrávání dat na server, což umožní výzkumníkům lehčí přístup k datům uživatelů a nebo pouze pro ukládání dat či stahování dat (cloud uložení). Dále pak mohou vzniknout různé jiné analýzy a jejich zobrazení, jelikož se v aplikaci prozatím používá pouze vizualizace časových řad a jednotlivých měření.

Pokud se aplikace bude nadále vyvíjet a zařízení Sensatio by vlastnilo více lidí tak věřím, že aplikace má komerční potenciál.

Literatura

1. CACIOPPO, John T; TASSINARY, Louis G; BERNTSON, Gary. Handbook of psychophysiology. In: Cambridge university press, 2007, s. 159–181.
2. BOUCSEIN, Wolfram. *Electrodermal activity*. Springer Science & Business Media, 2012.
3. MALČÍK, Martin; MIKLOŠIKOVÁ, Miroslava; ZEMČÍK, Tomáš. Development of Wearable Devices for Measurement of Multiple Physiological Variables and Evaluation of Emotions by Fingerprints and Population Hypotheses. 2021.
4. KHUMAR, SS; KAUR, Paramjit; KAUR, Jasjeet. A study of therapeutic effect of GSR biofeedback on mild hypertension. *Journal of the Indian Academy of Applied Psychology*. 1992.
5. NAGAI, Yoko; JONES, Christopher Iain; SEN, Arjune. Galvanic Skin Response (GSR)/Electrodermal/Skin Conductance Biofeedback on Epilepsy: A Systematic Review and Meta-Analysis. *Frontiers in Neurology*. 2019, roč. 10, s. 377. ISSN 1664-2295. Dostupné z DOI: 10.3389/fneur.2019.00377.
6. KOLARSICK, Paul A. J.; KOLARSICK, Maria Ann; GOODWIN, Carolyn. Anatomy and Physiology of the Skin. *Journal of the Dermatology Nurses' Association*. 2011, roč. 3, č. 4. ISSN 1945-760X. Dostupné také z: https://journals.lww.com/jdnaonline/Fulltext/2011/07000/Anatomy_and_Physiology_of_the_Skin.3.aspx. [cit. 2021-03-24].
7. EUCERIN. *Vše o Kůži: Struktura a Funkce Kůže: Úvod*. Dostupné také z: <https://www.eucerin.cz/o-kuzi/zakladni-informace/struktura-a-funkce-kuze>. [cit. 2021-03-24].
8. EBLING, F. John G.; MONTAGNA, William. Human skin. 2020. Dostupné také z: <https://www.britannica.com/science/human-skin>. [cit. 2021-03-23].
9. BROWN, Thomas M; KRISHNAMURTHY, Karthik. Histology, dermis. *StatPearls [Internet]*. 2020.
10. *Skin*. Encyclopædia Britannica, 2013. Dostupné také z: <https://www.britannica.com/science/human-skin#/media/1/547591/2027>. [cit. 2021-03-27].
11. PROKASY, William. *Electrodermal activity in psychological research*. Elsevier, 2012.
12. FINK, George. *Stress science: neuroendocrinology*. Academic Press, 2010.

13. FOWLES, Don C.; CHRISTIE, Margaret J.; EDELBERG, Robert; GRINGS, William W.; LYKKEN, David T.; VENABLES, Peter H. Publication Recommendations for Electrodermal Measurements. *Psychophysiology*. 1981, roč. 18, č. 3, s. 232–239. Dostupné z DOI: <https://doi.org/10.1111/j.1469-8986.1981.tb03024.x>. [cit. 2021-03-27].
14. PABST, Oliver; TRONSTAD, Christian; GRIMNES, Sverre; FOWLES, Don; MARTINSEN, Ørjan G. Comparison between the AC and DC measurement of electrodermal activity. *Psychophysiology*. 2017, roč. 54, č. 3, s. 374–385. Dostupné z DOI: <https://doi.org/10.1111/psyp.12803>. [cit. 2021-03-26].
15. CALLAHAM, John. *The history of Android: The evolution of the biggest mobile OS in the world*. Android Authority, 2020-11. Dostupné také z: <https://www.androidauthority.com/history-android-os-name-789433/>. [cit. 2021-04-17].
16. *Android OS version market share over time*. AppTornado GmbH, 2021-04. Dostupné také z: <https://www.appbrain.com/stats/top-android-sdk-versions>. [cit. 2021-04-26].
17. *Android - Architecture*. Tutorialspoint, 2012-12. Dostupné také z: https://www.tutorialspoint.com/android/android_architecture.htm. [cit. 2021-04-17].
18. PATIDAR, Sundar. *Android Architecture*. 2016-12. Dostupné také z: <https://spatidarblog.files.wordpress.com/2016/11/android-system-architecture.jpg>. [cit. 2021-04-25].
19. *Understand the Activity Lifecycle*. Google LLC, 2020-04. Dostupné také z: <https://developer.android.com/guide/components/activities/activity-lifecycle>. [cit. 2021-04-17].
20. GOOGLE. *Understand the Activity Lifecycle*. Google LLC, 2020-04. Dostupné také z: https://developer.android.com/guide/components/images/activity_lifecycle.png. [cit. 2021-04-20].
21. GOOGLE. *Processes and Application Lifecycle*. Google LLC, 2020-03. Dostupné také z: <https://developer.android.com/guide/components/activities/process-lifecycle>. [cit. 2021-04-17].
22. *Android Directory Structure*. CodePath, 2014-08. Dostupné také z: <https://guides.codepath.com/android/Android-Directory-Structure>. [cit. 2021-04-17].
23. GUPTA, Naresh Kumar. *Inside Bluetooth low energy*. Artech House, 2016.
24. GOOGLE. *Meet Android Studio*. Google LLC, 2021-02. Dostupné také z: <https://developer.android.com/studio/intro>. [cit. 2021-04-06].
25. ORACLE. *Java™ Platform Standard Edition 16 Development Kit*. 2017-09. Dostupné také z: <https://www.oracle.com/java/technologies/javase/jdk16-readme-downloads.html>. [cit. 2021-04-09].

26. DIMARZIO, Jerome. *ANDROID A PROGRAMMERS GUIDE*. 1. vyd. USA: McGraw-Hill, Inc., 2008. ISBN 0071599886.
27. GOOGLE. *Create and manage virtual devices*. Google LLC. Dostupné také z: <https://developer.android.com/studio/run/managing-avds>. [cit. 2021-04-24].
28. HOFFMAN, Chris. *What Is a CSV File, and How Do I Open It?* How-To Geek, 2018-04. Dostupné také z: <https://www.howtogeek.com/348960/what-is-a-csv-file-and-how-do-i-open-it/>. [cit. 2021-04-09].
29. TOWNSEND, Kevin. *Introduction to Bluetooth Low Energy*. Adafruit IO, 2014-03. Dostupné také z: <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gap>. [cit. 2021-04-09].
30. TOWNSEND, Kevin. *Introduction to Bluetooth Low Energy*. Adafruit IO, 2014-03. Dostupné také z: <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>. [cit. 2021-04-09].
31. GOOGLE. *Bluetooth low energy*. Google LLC. Dostupné také z: <https://developer.android.com/guide/topics/connectivity/bluetooth-le>. [cit. 2021-04-24].
32. BAUER, Vladislav. *A categorized directory of libraries and tools for Android*. 2016-06. Dostupné také z: <https://android-arsenal.com/tag/40>. [cit. 2021-04-13].
33. GEHRING, Jonas. *jjoe64/GraphView*. GitHub, 2013-08. Dostupné také z: <https://github.com/jjoe64/GraphView>. [cit. 2021-04-13].
34. ALEXANDROS, SCHILLINGS. *Bluetooth LE Scanner*. Google LLC, 2020-06-19. Ver. 1.1.39. Dostupné také z: <https://play.google.com/store/apps/details?id=uk.co.alt236.btlescan&hl=en&gl=US>. [cit. 2021-04-26].
35. *Introduction to Google Drive API*. Google, 2020-12. Dostupné také z: <https://developers.google.com/drive/api/v3/about-sdk>. [cit. 2021-04-27].

Příloha A

Obsah přílohy

Následující části byly vloženy do přílohy v IS Edison.

A.1 Zdrojový kód aplikace

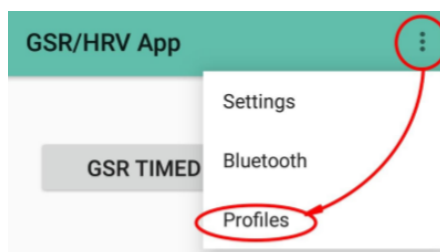
A.2 Instalační soubor .apk

Příloha B

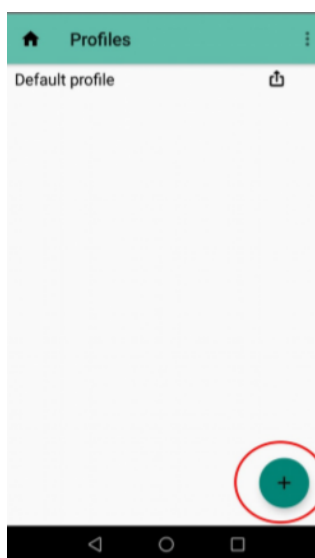
Příručka

B.1 Tvorba profilu

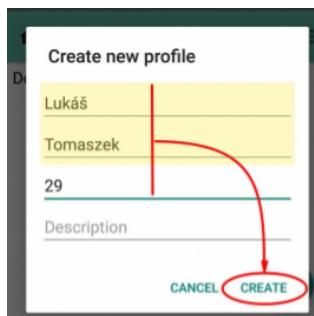
1. V pravém horním rohu stikněte symbol se **třemi tečkami** a následně zvolte položku **Profi-les/Profily**.



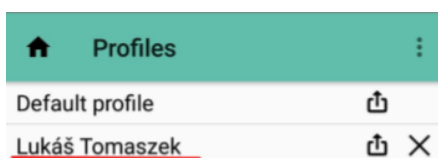
2. V pravém dolním rohu stiskněte kolečko se symbolem +.



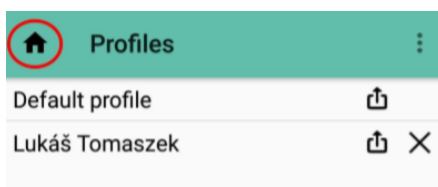
3. Vyplňte potřebné údaje a potvrďte stisknutím **Create/Vytvořit**.



4. Váš profil se vytvořil a nyní jej vidíte v seznamu profilů. V případě tvorby více profilů opakujte kroky 2 – 4.

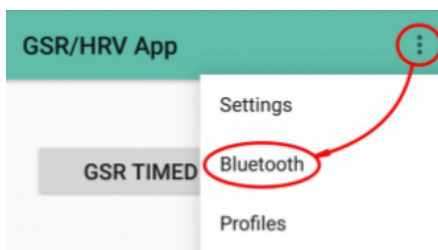


5. Na hlavní obrazovku se vrátíte stiskem ikony domečku v levé horní části obrazovky.

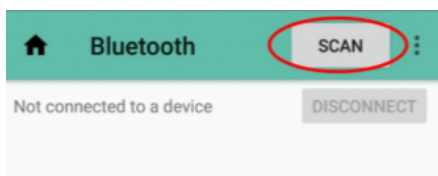


B.2 Připojení zařízení

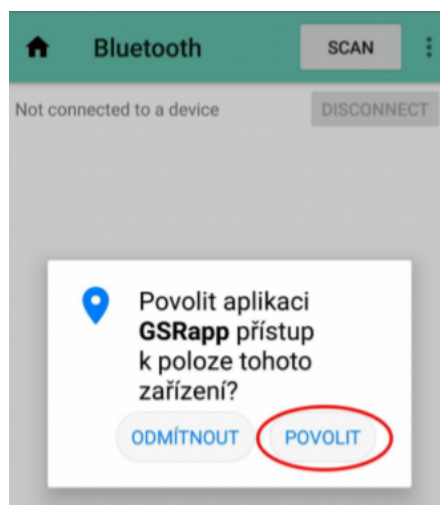
1. V pravém horním rohu stikněte symbol se třemi tečkami a následně zvolte položku **Bluetooth**.



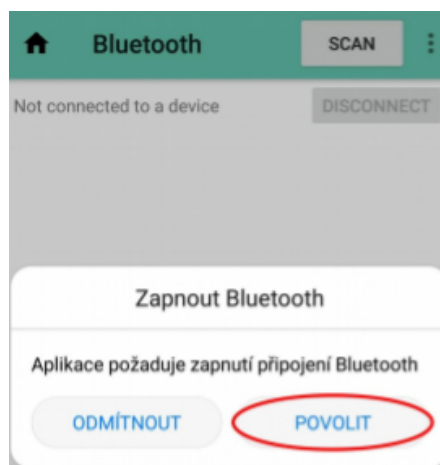
2. Stiskněte tlačítko **Scan/Skenovat** v pravém horním rohu.



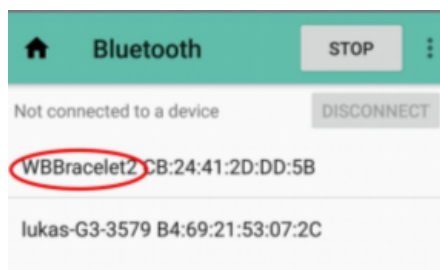
- (a) V případě prvního vyhledávání po Vás aplikace bude vyžadovat práva (poloha, Bluetooth, ...). Jednotlivá práva se liší podle verze Android a tato práva slouží výhradně k vyhledávání náramku a čtení dat z něj.



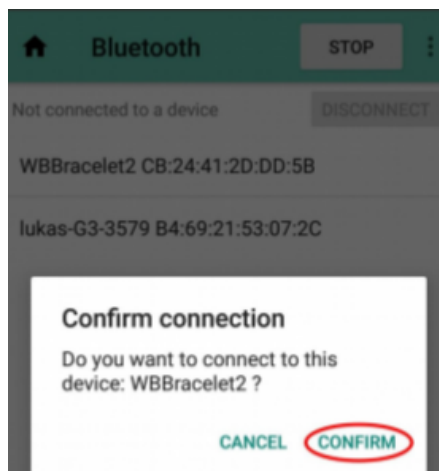
- (b) V případě vypnutého Bluetooth na Vašem zařízení bude aplikace požadovat jeho zapnutí.



3. Vyberte náramek, ke kterému se má aplikace připojit. Jedná se o zařízení s názvem **WBBracelet2**. Výběr provedete stiskem na požadované zařízení.



4. Potvrďte Vaši volbu stiskem tlačítka **Confirm/Potvrdit**.



5. Náramek se připojí k Vašemu zařízení. Zobrazí se Vám hláška v dolní části obrazovky a zároveň na prvním řádku uvidíte, k jakému zařízení jste připojeni.

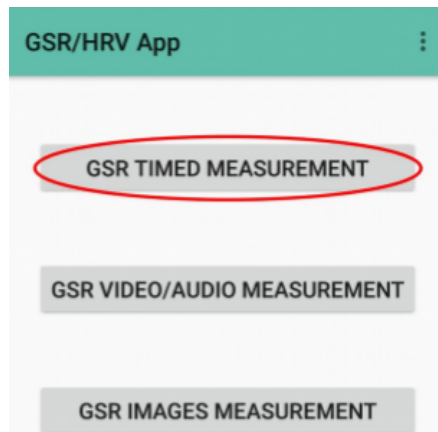


- (a) V případě nepřipojení opakujte kroky 3-4.

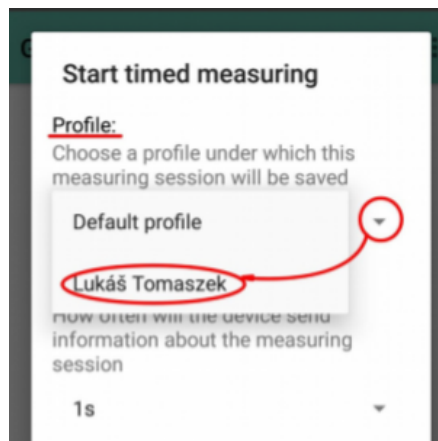
6. Na hlavní obrazovku se vrátíte stiskem ikony domečku v levé horní části obrazovky.

B.3 Připojení zařízení

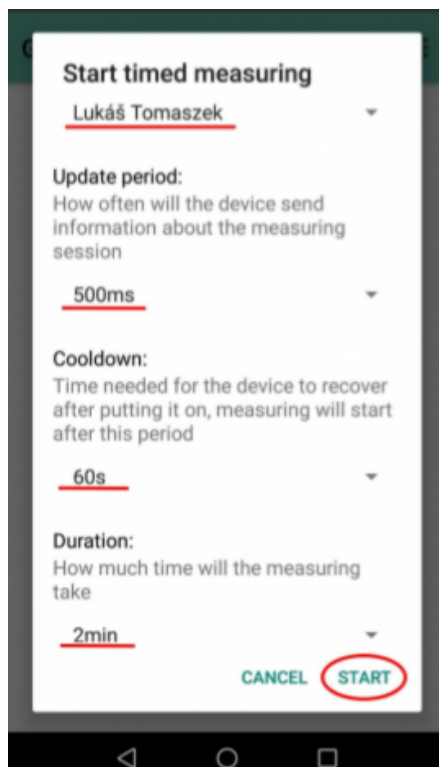
1. Připojte zařízení.
2. Stiskněte tlačítko **GSR Timed Measurement/GSR časové měření**.



3. Nastavte **Profile/Profil** na **Váš profil**.



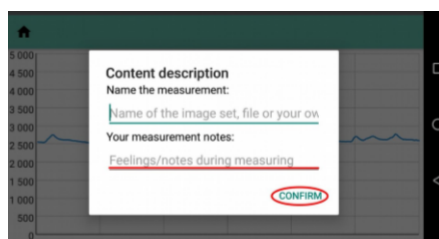
4. Nastavte **Update period/Obnovovací rychlost** na **500ms**.
5. Nastavte **Cooldown/Zotavovací dobu** na **60s**.
6. Nastavte **Duration/Doba trvání** na **2min**.
7. Výsledné nastavení pak vypadá následovně. Potvrďte stisknutím tlačítka **Start**.



8. Nyní probíhá měření.



9. Po uplynutí 3min se ozve zvukový signál, který značí konec měření. Můžete vyplnit poznámky k danému měření a potvrdit stiskem **Confirm/Potvrdit**.



10. Po měření se zobrazí přehled měření.

(a) Měření **bylo timed/časované** a lze si prohlédnout statistiky a graf měření, stejně jak v normálním přehledu měření.

(b) Měření **nebylo timed/časované** a lze si prohlédnout graf z měření

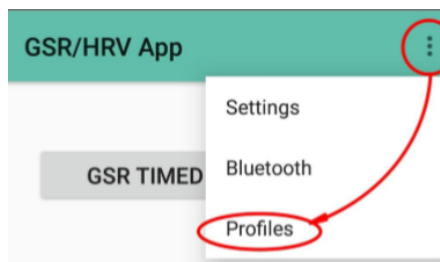
11. Na hlavní obrazovku se vrátíte stiskem ikony domečku v levé horní části obrazovky.

B.3.1 Dodatky k měření

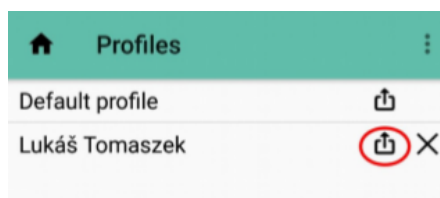
- V aplikaci existuje více měření, každé z nich si můžete vyzkoušet. Proces nastavování a měření funguje na stejném principu.
- U měření s obrázky či videem/zvukem je při nastavování nutno zvolit soubory k přehrání během měření.
- U měření **infinite/nekonečného** lze měření po nastavení spustit, pozastavit a ukončit tlačítky **start/začít**, **pause/pauza** a **stop/zastavit**.

B.4 Export dat z daného profilu

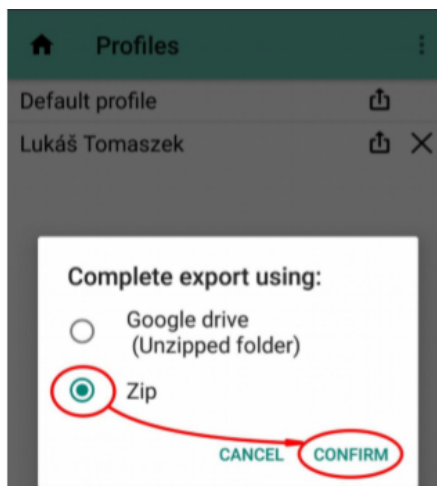
1. V pravém horním rohu stikněte symbol se **třemi tečkami** a následně zvolte položku Profiles.



2. U profilu, který chcete exportovat stiskněte ikonku exportu.



3. Zvolte možnost **zip** a potvrďte tlačítkem **Confirm/Potvrdit**.



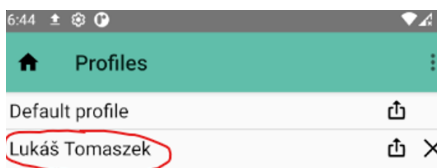
(a) Vyberte variantu Google drive (postupujte podle pokynů).

4. Na hlavní obrazovku se vrátíte stiskem ikony domečku v levé horní části obrazovky.

B.5 Přehled měření

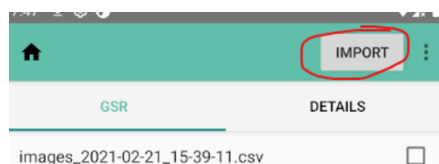
1. Navigujte do profilů (tři tečky na liště -> **Profiles/Profilý**).

2. Vyberte profil, který chcete zobrazit.

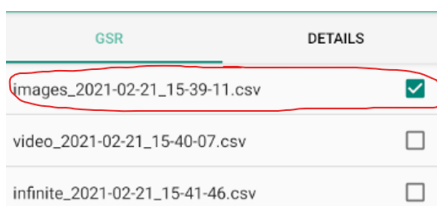


3. Pokud jste již něco naměřili na tomto profilu, tak by zde měly být měření.

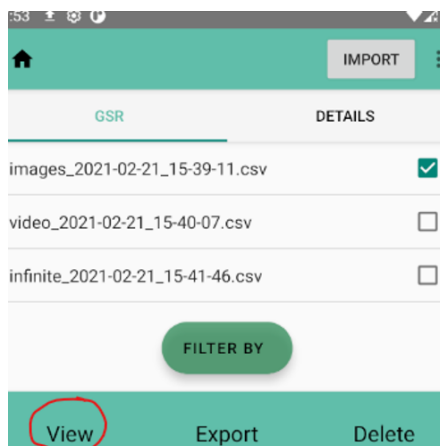
(a) Můžete importovat měření na vrchní části stránky (postupuje podle pokynů).



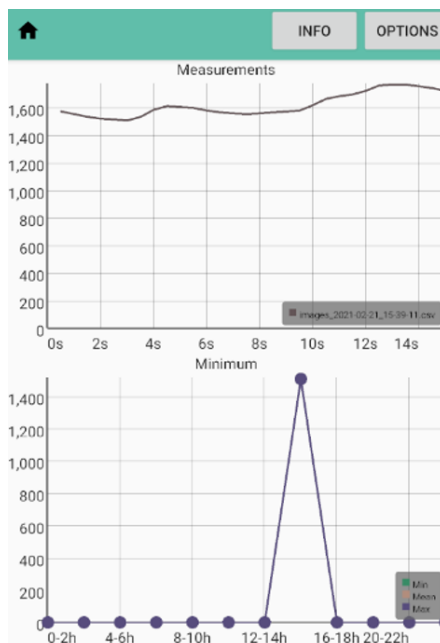
4. Vyberte měření kliknutím na ně.



- (a) Měření se dají i vyfiltrovat po stisknutí tlačítka **filter by/filtrovat podle**.
5. Po vybrání měření se zobrazí lišta na spodní části stránky, kde si můžete vybrat, zda-li chcete vybraná měření smazat, exportovat či zobrazit. Pro zobrazení vybraných měření zmáčknete **view/přehled**.



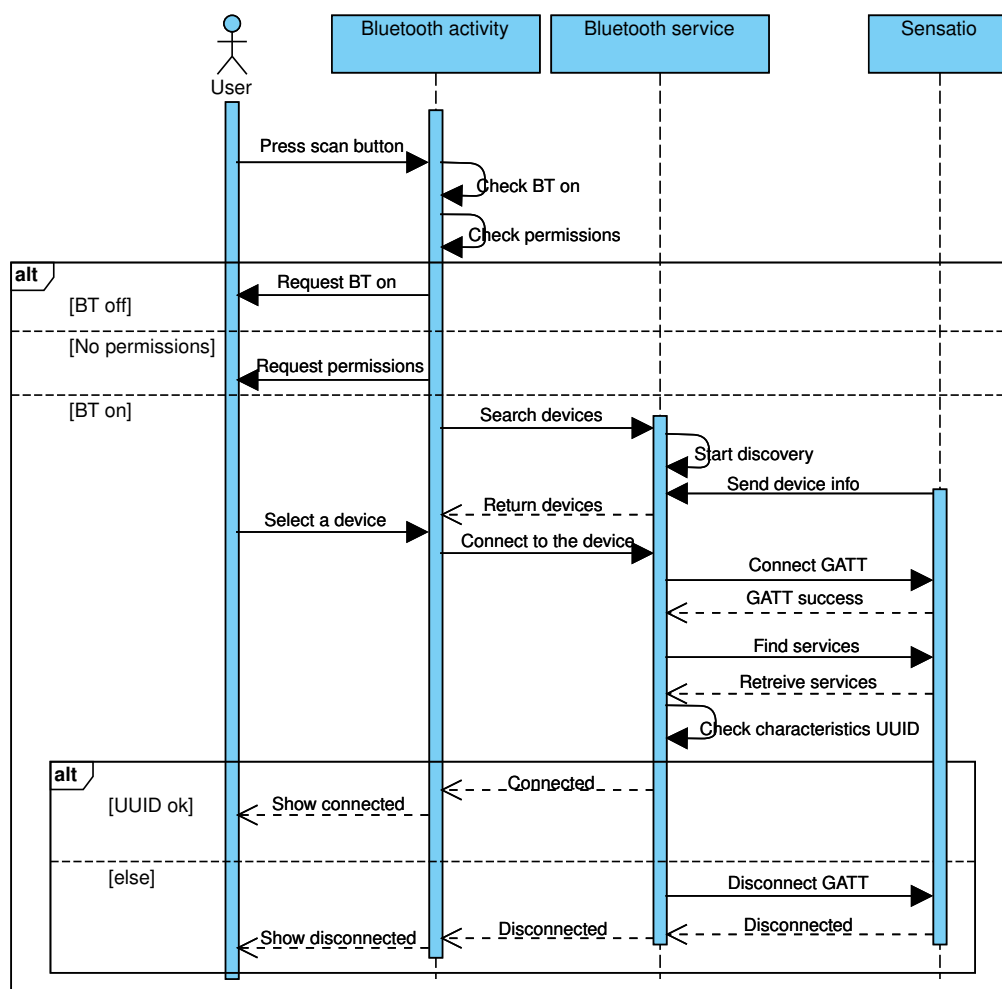
6. Vybraná měření jsou ukázány na grafu. Pod tímto grafem jsou i analýzy, složené z vybraných měření.



7. Na vrcholu stránky jsou tlačítka pro ukázání informací o všech měřeních (tlačítko **info**) a pro práci s grafy (tlačítko **options/možnosti**).
8. Na hlavní obrazovku se vrátíte stiskem ikony domečku v levé horní části obrazovky.

Příloha C

Sekvenční diagram připojení bluetooth



Obrázek C.1: Sekvenční diagram připojení měřicího zařízení k mobilnímu přes Bluetooth